HS-ES-DE: HS-ES Followed by L-SHADE-EpSin for Real Parameter Single Objective Optimization

Jianjun Jiang[a], Yong Wang[a,b], Chengjun Li[a,b,*], Chuchuan Cen[a], Yikai Shao[a], Qinxue Meng[c]

[a] *School of Computer Science, China University of Geosciences, Wuhan 430074, P.R. China*
[b] *Hubei Key Laboratory of Intelligent Geo-Information Processin, China University of Geosciences, Wuhan 430074, P.R. China*
[c] *School of Software, University of Technology Sydney, Sydney 2000, Australia*

Abstract

For real parameter single objective optimization, Differential Evolution (DE) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES) both perform powerfully. Nevertheless, in the field of real parameter single objective optimization, it is impossible for a given algorithm to perform well in all fitness landscapes. Practice has proved that ensemble of different algorithms may lead to improvement in solution. In this paper, based on two famous population-based metaheuristics - L-SHADE-EpSin and HS-ES, we propose ensemble with successively executed constituent algorithms - HS-ES-DE. In our algorithm, HS-ES is replaced by L-SHADE-EpSin after stagnation is detected. Beside our HS-ES-DE, 12 population-based metaheuristics are involved in our experiments in which three benchmark test suites are employed. Experimental results show that our algorithm is very competitive.

Keywords: Real parameter single objective optimization, population-based mataheuristic, ensemble, stagnation

1. Introduction

Real parameter single objective optimization aims to find the best decision vector in solution space to minimize (or maximize) an objective function which is a research focus in artificial intelligence. So far, a variety of population-based metaheuristics have been proposed in this field. Top evolutionary computation venue, Congress of Evolutionary Computation (CEC), held at least six

*Corresponding author

competitions on real parameter single objective optimization among population-based metaheuristics annually.

The CEC competitions are famous in the research of real parameter single objective optimization and have a great impact on the future research directions in this field. We briefly introduce the winners of each of the competitions as follows.

- NBIPOP-aCMA-ES [10], the winner in 2013, is a variant of Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [6] with a restart scheme including two regimes;

- L-SHADE [17], the winner in 2014, a variant of Differential Evolution (DE) [15], is extended from SHADE [16] with the Linear Population Size Reduction (LPSR).

- UMOEAs-II [5], a joint winner in 2016, is ensemble of DE and CMA-ES.

- L-SHADE-EpSin [2], the other joint winner in 2016, is a variant of L-SHADE with adaptive parameter settings and the local search based on Gaussian walks.

- EBOwithCMAR [8], the winner in 2017, is a self-adaptive butterfly optimizer with the success history based adaption, the LPSR, and the covariance matrix adapted retreat phase;

- HS-ES [22], the winner in 2018, is controlled by CMA-ES with a restart scheme and an improved version of the univariate sampling.

- IMODE [12], the winner in 2020, is a DE algorithm with three DE mutation strategies and the sequential quadratic programming.

Among these seven winners, NBIPOP-aCMA-ES and HS-ES are based on CMA-ES, while L-SHADE, L-SHADE-EpSin, and IMODE are based on DE. Moreover, UMOEAs-II is based on both CMA-ES and DE. In [14], further comparison shows that UMOEA-II, L-SHADE-EpSin, and HS-ES are top performers among the six winners in the previous five competitions. This means that the top three performers in the previous five competitions are based on DE and/or CMA-ES. Thus, it can be seen according to the CEC competitions that both CMA-ES and DE perform better in real parameter single objective optimization than other approaches. DE and CMA-ES were initially proposed in [15] and [7] respectively. In the population of DE, operators such as mutation, crossover and selection, are exerted on individuals, i.e., target vectors. In the evolutionary framework of DE, trail vectors for next generation are produced via mutation firstly and then crossover which are

specified as trail vector generation strategies. In the stage of selection, a target vector or its trial vector is chosen based on fitness comparison to be a target vector of the next generation. In population of CMA-ES, individual center is the key to generate next generation. Specifically, individual center is initialized by the part of individuals with better fitness than the others. Then, new individuals are generated by sampling from a Gaussian distribution based on the individual center. After that, individual center of the next generation is decided. The motivation of this paper is given below. It has be proven that CMA-ES is easy to fall into stagnation, the state means existing difference among individuals not enough to support valid search any more. Therefore, in both NBIPOP-aCMA-ES and HS-ES, restart is employed to enhance CMA-ES. Furthermore, in HS-ES, the improved version of univariate sampling is called after the step of CMA-ES for further enhancement. In fact, compared with restart and the univariate sampling, it may be more useful that another powerful population-based metaheuristic takes over search after stagnation of CMA-ES. Based on the idea, we can obtain ensemble with successively executed constituent algorithms.

In this paper, we propose HS-ES-DE, an ensemble of HE-ES and L-SHADE-EpSin. Both of the constituent algorithms are winners in the CEC competitions. Our proposed algorithm starts with HS-ES where a scheme is applied to detect stagnation. In detail, stagnation is confirmed provided that improving ratio of the average fitness from the previous generation to the current one is lower than a threshold for a given number of contiguous generations. After stagnation detected, HS-ES hand over the evolutionary procedure to L-SHADE-EpSin. For taking over, population size and population diversity are both adjusted by our scheme. The initial value of population size of L-SHADE-EpSin - NP_DE_{max} - is calculated based on the rest number of function evaluations before taking over. Then, in the last generation for the step of CMA-ES, NP_DE_{max} individuals are produced. After that, mutation and crossover in the original version of DE [15] is executed for a number of generations. In the generations, offspring produced by the trial vector generation strategy are all directly selected without considering fitness to improving diversity.

We evaluate our proposed algorithm on the benchmark test suites of CEC 2014, 2017, and 2020 for real parameter single objective optimization. In the first experiment based on the CEC 2014 benchmark test suite, we compare our HS-ES-DE with L-SHADE-Epsin, HS-ES, and other seven population-based metaheuristics when dimensionality are set 30, 50, and 100, respectively. In the second experiment based on the CEC 2017 benchmark test suite, we compare HS-ES-DE with the same algorithms when dimensionality are set 30. In the third experiment based on the CEC

2020 benchmark test suite, we compare our algorithm with the top three algorithms in the 2020 competitions when dimensionality are set 20. The experimental results show that our algorithm outperforms other population-based metaheuristics for real parameter single objective optimization.

The rest of this paper is organized as follows. We firstly reviewed related work in Section II. Section III explains the details of our proposed algorithm. Experimental results are demonstrated and discussed in Section IV with a conclusion in Section V.

2. Related Work

In this section, we reviewed recent research on population-based metaheuristics for real parameter single objective optimization in an ensemble way covering ensembled DE algorithms and ensembled population-based metaheuristics.

Wu [20] proposed MPEDE which simultaneously consists of three mutation strategies. There are three equal-sized smaller indicator subpopulations and one much larger reward subpopulation in population of MPEDE. Each constituent mutation strategy controls one indicator subpopulation. The current best performing mutation strategy is determined according to the ratio between fitness improvement and consumed function evaluations every a certain number of generations. Then the reward subpopulation is allocated to the determined best performing mutation strategy dynamically. Mostafa et al. [1] presented sTDE-dR whose population is clustered in multiple tribes and utilizes an ensemble of different mutation and crossover strategies. Both the life cycle and participation ratio for the next generation are determined by a success-based scheme. In each tribe, scaling factor and crossover rate are controlled by a different adaptive scheme. The mean success of each tribe is used to calculate the participation ratio for the next generation. Population size is dynamically reduced during execution. Sallam et al. [13] proposed a DE algorithm considering both fitness landscapes and performance history of the operators to dynamically selecting the most suitable operator. Cui et al. [3] presented AMECoDEs which employs two elites-guided trial vector generation strategies for each individual to generate two candidate solutions accordingly. In fact, only the better one can participate in selection. Wu et al. [21] proposed EDEV which consists of three highly popular and efficient DE variants - JADE [23], CoDE [19], and EPSDE. Similar with MPEDE, there are three indicator subpopulations and a reward subpopulation in population. Also, the competition mechanism for the four subpopulations are similar with that in MPEDE. Zhang et al. [24] suggested MLCC framework which implements a parallel structure with the entire population simultaneously

monitored by multiple DE algorithms assigned to different layers. An individual can store, utilize and update its evolution information in different layers. MLCC-SI is regarded as the representative of MLCC-based DE algorithms.

Elsayed et al. [5] proposed UMOEAs-II which employ a DE algorithm with multiple combinations of operators and CMA-ES. Each of the two constituent algorithms controls a subpopulation. The two subpopulation shares information at intervals. Liu et al. [9] presented CETMS based on tissue membrane systems and CMA-ES. In the framework of the tissue-like membrane system, network structure consists of several of cells. Each cell has its own multisets of objects. CMA-ES is employed as the reaction rules to evolve objects in different cells. Mohamed et al. [11] suggested L-SHADE-SPACMA in which individuals are controlled by either L-SHADE-SPA or a modified version of CMA-ES. Better performer can control more individuals.

3. Methodology

In this section, we present the details of HS-ES and L-SHADE-EpSin firstly. Then, based on our analysis on the two algorithms, we introduce our algorithm - HS-ES-DE. The pseudo-code of all the three algorithms is given here.

3.1. The involved two population-based metaheuristic

3.1.1. HS-ES

The pseudo-code of HS-ES is in Algorithm 1. In essence, HS-ES is ensemble of the improved version of the univariate sampling [22] and CMA-ES [6]. Here, the two constituent algorithms are not further explained. Details of them can be found in [22] and [6].

3.1.2. L-SHADE-EpSin

The pseudo-code of L-SHADE-EpSin is given in Algorithm 2 followed by the equations required further explanation. Nevertheless, mutation, crossover, and selection, which are all inherited from JADE [23], are not detailedly explained here.

In the first stage of L-SHADE-EpSin, the non-adaptive sinusoidal decreasing adjustment for each target vector in the gth generation is

$$F_{i,g} = \frac{1}{2} \cdot \left(\sin\left(2\pi \cdot freq \cdot g + \pi\right) \cdot \frac{G_{\max} - g}{G_{\max}} + 1 \right), \tag{1}$$

Algorithm 1 The pseudo-code of HS-ES

Input: NP_1, population size in the first step based on the univariate sampling; NP_2, population size of CMA-ES (variable); NP_3, population size of the final step based on the univariate sampling; G, the maximum number of generations for the first step of the univariate sampling; T, the maximum times of CMA-ES execution; Thr, threshold; $MaxFES$, the maximum number of fitness evaluations

Parameter: t, g, S_1, S_2, FES

$g = 0$, $FES = 0$
Initialize
while $g < G$ **do**
    Implement the improved version of the univariate sampling on a randomly selected dimension
    $g = g + 1$
end while
$FES = FES + G \cdot NP_1$
$S_1 \leftarrow$ the best solution
while $t < T$ **do**
    Regard S_1 as the beginning of CMA-ES
    repeat
        Implement CMA-ES for a generation
        $FES = FES + NP_2$
    until the difference between the current best solution and the best solution in the previous generation is less than Thr
end while
$S_2 \leftarrow$ the best solution in the step of CMA-ES
while $FES < MaxFES$ **do**
    Detect which dimension of S_2 needed be fixed
    Fix the detected dimension with the univariate sampling
    $FES = FES + NP_3$
    $S_2 \leftarrow$ the best solution
end while
Report S_2

Algorithm 2 The pseudo-code of L-SHADE-EpSin

Input: NP_{max}, the maximum value of population size; NP_{min}, the minimum value of population size; $MaxFES$, the maximum number of fitness evaluations

Parameter: S_{freq}, S_F, S_{CR}, μF, μCr, and, $\mu Freq$, FES

$\mu F = 0.5$, $\mu Cr = 0.5$, $\mu Freq = 0.5$, $FES = 0$, $g = 1$, and $NP = NP_{max}$
while $FES <= MaxFES$ **do**
  if $FES <= 0.5 \cdot MaxFES$ **then**
    for $i = 1$ to NP **do**
      Generate a random number $c = rand(0, 1)$
      if $c < 0.5$ **then**
        Generate $F_{i,g}$ using the non-adaptive decreasing adjustment shown in Equation 1
      else
        Generate $F_{i,g}$ using the adaptive increasing adjustment shown in Equation 2
      end if
      Generate $Cr_{i,g}$ based on Equation 5
    end for
  else
    $S_F, S_{CR} = \phi$
    for $i = 1$ to NP **do**
      Generate a random index $r_i = rand(1, H)$
      Generate $F_{i,g}$ and $Cr_{i,g}$ based on Equations 4 and 5, respectively
    end for
  end if
  for $i = 1$ to NP **do**
    Generate $p_i = rand(0, 1) \cdot N$, where $N = 0.1 \cdot NP$
    Mutation based on DE/current-to-pbest/1 with the optional external archive
    Binomial crossover
    Selection
    Store successful $freq_{i,g}$, $F_{i,g}$, and $Cr_{i,g}$ in S_freq, S_F, and S_{Cr}, respectively
  end for
  Compute $\mu freq_{r_i,g+1}$, $\mu F_{r_i,g+1}$, and $\mu Cr_{r_i,g+1}$ based on Equations 6, 7, and 8, respectively
  Call the LPSR according to Equation 12
  if $NP \leq 20$ for the first time **then**
    Call the local search based on Gaussian Walks based on Equation 13
  end if
  g=g+1
end while
Report solution

where i is from 1 to NP, g is the current generation number, $freq$ represents the frequency of sinusoidal function, and G_{max} is the maximum generation, which is computed based on the maximum number of function evaluations $MaxFES$ and the settings of the LPSR. Meanwhile, the adaptive sinusoidal increasing adjustment for each target vector in the gth generation is

$$F_{i,g} = \frac{1}{2} \cdot \left(\sin\left(2\pi \cdot freq_{i,g} \cdot g\right) \cdot \frac{g}{G_{\max}} + 1 \right), \tag{2}$$

where $freq_{i,g}$ is computed based on Equation 3.

$$freq_{i,g} = randc\left(\mu freq_{r_i}, 0.1\right) \tag{3}$$

In Equation 3, $\mu freq_{r_i}$ is chosen randomly from the external memory M_{freq} which stores the successful mean frequencies in history. In the second stage,

$$F_{i,g} = randc\left(\mu F_{r_i}, 0.1\right), \tag{4}$$

where μF_{r_i} is chosen randomly from the external memory M_F which stores the successful mean of F in history. During the whole course,

$$Cr_{i,g} = randn(\mu Cr_{r_i}, 0.1), \tag{5}$$

where μCr_{r_i} is chosen randomly from the external memory M_{Cr} which stores the successful mean of Cr in history. For update,

$$\mu freq_{r_i,g+1} = mean_L\left(S_{freq}\right), \tag{6}$$

$$\mu F_{r_i,g+1} = mean_L\left(S_F\right), \tag{7}$$

$$\mu CR_{r_i,g+1} = mean_L\left(S_{CR}\right). \tag{8}$$

Here, $mean_L(S_{freq})$, $mean_L(S_F)$, and $mean_L(S_{CR})$ are all computed based on the following equations,

$$mean_L(S) = \frac{\sum_{i=1}^{|S|} w_i \cdot S_i^2}{\sum_{i=1}^{|S|} w_i \cdot S_i}, \tag{9}$$

$$w_i = \frac{\Delta f_i}{\sum_{j=1}^{|S|} \Delta f_j}, \tag{10}$$

$$\Delta f_i = |f(u_{i,g}) - f(x_{i,g})|. \tag{11}$$

For the LPSR,

$$NP = Round[\frac{NP_{min} - NP_{max}}{MaxFES} \cdot FES + NP_{max}] \tag{12}$$

For the Gaussian Walks, ten randomly selected target vectors are reinitialized. After that, in the following G_{ls} generations,

$$y_m = Gaussian(\mu_b, \sigma) + (\epsilon \cdot \vec{x}_{mbest} - \hat{\epsilon} \cdot x_m), \tag{13}$$

where $m \in [1, 10]$ ϵ and $\hat{\epsilon}$ are two uniform random numbers in the range [0,1], and $\vec{x}_{mbest}$ is the best individual among the reinitialized ten ones. Here,

$$\sigma = \left| \frac{\log(G)}{G} \cdot (\vec{x}_m - \vec{x}_{mbest}) \right| \tag{14}$$

3.2. Our algorithm

The first step of HS-ES for the univariate sampling can be regarded as a initialization method better than randomization. In the second step, CMA-ES may run more than one time. However, the termination of each CMA-ES execution may be long before stagnation, because the criterion for termination - no progress on best solution in just one generation - cannot reflect stagnation in most occasions. That is, in HS-ES, CMA-ES is employed even repeatedly to just obtain approximate solution. After that, the univariate sampling is expected for further progressing based on the foundation made by CMA-ES.

We believe that the first step for initialization is reasonable. Nevertheless, it may be a better choice that CMA-ES is executed until stagnation occurs. After all, CMA-ES is powerful. To make use of the remaining fitness evaluations after stagnation, another population-based metaheuristic may take over population. Here, we choice L-SHADE-EpSin. Reasons are given below. Firstly, L-SHADE-EpSin can keep search ability longer than many other population-based metaheuristics. The fact can be reflected in our following experiments. In addition, as a DE algorithm with the LPSR, L-SHADE-EpSin cost much less fitness evaluations in generations of the later stage than in those of the earlier stage.

Considering these issues, we propose HS-ES-DE shown in Algorithm 3. Improving rate of average fitness IR in Step 12 is calculated as

$$IR = \frac{|\overline{f}_p - \overline{f}_c|}{\overline{f}_p}, \tag{15}$$

where $\overline{f}_p$ denotes the average fitness in the previous generation and $\overline{f}_c$ denotes the average fitness in the current generation. It can be seen from Step 12 that we use a very strict criterion to detect stagnation.

Algorithm 3 The pseudo-code of HS-ES-DE

Input: NP_1, population size in the first step based on univariate sampling; NP_2, population size of CMA-ES (variable); NP_DE, population size of L-SHADE-EpSin (variable); G, Maximum generations for the first step of univariate sampling; Thr, threshold; Gen_1, number of generations for detecting improving rate of average fitness; Gen_2, number of generations for executing the original version of DE mutation and crossover; $MaxFES$, the maximum number of fitness evaluations

Parameter: g, S_1, FES

$g = 0$, $FES = 0$
Initialize
while $g < G$ **do**
    Implement the improved version of univariate sampling on randomly selected dimension
    $g = g + 1$
end while
$FES = FES + G \cdot NP_1$
Regard S_1, the best solution, as the beginning of the step of CMA-ES
repeat
    Implement CMA-ES for a generation
    $FES = FES + NP_2$
until improving rate of average fitness, which is shown in Equation 15, has been lower than Thr in contiguous Gen_1 generations
Execute CMA-ES one more generation without fitness evaluation to obtain NP_DE_{init} counted based on Equation 12
$g = 0$
while $g < Gen_2$ **do**
    Execute the original version of DE mutation and crossover
    Directly select offspring without fitness evaluation
end while
while $FES < MaxFES$ **do**
    Execute L-SHADE-EpSin
    $FES = FES + NP_3$
end while
Report solution

To produce the right number of individuals for L-SHADE-EpSin, CMA-ES is executed without fitness evaluation in one more generation after stagnation is confirmed. Then, individuals are diversified based on DE with directly selecting offspring. Similarly, fitness evaluation is not required here. In short, our processing scheme on population for taking over requires no fitness evaluation.

4. Experiments

In our experiments, we compare our HS-ES-DE with 12 population-based metaheuristics based on the CEC 2014, 2017, and 2020 benchmark test suites. In the three suites, functions can be divided into four types as shown in Table 1. In our comparison based on the CEC 2014 and

Table 1: Classification of the functions in the CEC 2014 and CEC 2017 benchmark test suites

Type	Function		
	2014	2017	2020
Unimodal	F1-F3	F1-F3	F1
Simple multimodal	F4-F16	F4-F10	F2-F4
Hybrid	F17-F22	F11-F20	F5-F7
Composition	F23-F30	F21-F30	F8-F10

2017 suites, L-SHADE-EpSin, UMOEAs-II, MPEDE, ETI-JADE [4], HS-ES, AMECoDEs, EDEV, MLCC-SI, and NDE [18] are selected as peers. That is, the top three performers in the CEC competitions and some up-to-date algorithms are involved. It requires to be emphasized that the maximum number of fitness evaluations is significantly increased in the CEC 2020 competition on real parameter single objective optimization. The top three algorithms in the competition with the new rule are IMODE, AGSK, and j2020. In our comparison based on CEC the 2020 suite, peers are just the above three population-based metaheuristics.

4.1. Experimental setting

Settings of all the algorithms are shown in Table 2, where D denotes dimensionality. In fact, in our HS-ES-DE, just the listed parameters need be considered. The value of other parameters coming from HS-ES and L-SHADE-EpSin is still kept by us.

Table 2: Settings of the involved algorithms

Algorithm	Parameters
L-SHADE-EpSin	$NP_{max} = D \cdot 18$, $NP_{min} = 4$, $\lvert A\rvert = 1.4 \times NP$, $H = 5$, $freq = 0.5$, and $G_{ls} = 250$ [2]
UMOEAs-II	$NP_{max} = D \cdot 18 + 4 + \lfloor log(D) \cdot 3\rfloor$, $NP_{min} = 8 + \lfloor log(D) \cdot 3\rfloor$, $prob_{ls} = 0.1$, and $cfe_{LS} = D \cdot 2000$ [5]
MPEDE	$NP = 250$, $\lambda_1 = \lambda_2 = \lambda_3 = 0.2$, and $ng = 20$ [20]
ETI-JADE	$NP = 100$, $\mu_F = 0.5$, $\mu_{CR} = 0.5$, $c = 0.1$, $\lvert A\rvert = 100$, and $p = 0.05$ [4]
HS-ES	$NP_1 = 200$, $NP_2 = 80 + \lfloor ln(D) \cdot 3\rfloor$, $NP_3 = 450$ when $D = 50$, $NP_3 = 600$ when $D = 100$, $cc = 0.96$, $I = 20$, $N^{Step1} = 100$, $N^{Step4} = 360$ when $D = 50$, and $N^{Step4} = 480$ when $D = 100$ [22]
AMECoDEs	$NP = 200$, $p = 0.1$, $c = 0.1$, $\epsilon = 0.001$, and $\lvert A\rvert = 200$ [3]
EDEV	$NP = 50$, $\lambda_1 = \lambda_2 = \lambda_3 = 0.1$, $\lambda_1 = 0.7$, and $ng = 20$ [21]
MLCC-SI	$NP = D \cdot 5$, $N = 0.05$ [24]
NDE	$NP_{max} = 300$, $NP_{min} = 5$, $gm = 10$, and $c = 0.1$ [18]
IMODE	$NP_{max} = D^2 \cdot 6$, $NP_{min} = 4$, $A = 2.6$, $H = D \cdot 20$, and $FES_{LS} = MaxFES \cdot 0.85$
AGSK	$NP_{max} = D \cdot 20$, $NP_{min} = 12$, $p = 0.05$, and $c = 0.05$
j2020	$bNP = D \cdot 7$, $sNP = D$, $\tau_1 = 0.1$, $\tau_2 = 0.1$, $ageLmt = \frac{MaxFES}{10}$, $eps = 1e-16$, $myEqs = 25$
HS-ES-DE	$Gen_1 = 50$, $Gen_2 = 4$, $Thr = 1E-11$, $F = 0.2$ (for mutation of the original version of DE), $Cr = 0.5$ (for crossover of the original version of DE)

In our experiments based on CEC 2014 and 2017 suites, the maximal number of function evaluations $MaxFES$ is set $D \cdot 10000$, i.e., 3.0E+05, 5.0E+05 and 1.0E+06 when $D = 30$, $D = 50$ and $D = 100$, respectively, based on the rule of the previous CEC competitions. In our experiment based on CEC 2020 suites, $MaxFES$ is set to 2.0E+05 and 1.0E+07, respectively, when $D = 20$. The former value is set based on the rule of the previous CEC competitions, while the latter value is set based on the rule of the CEC 2020 competition. For comparison, algorithm is always executed 30 times in each case.

4.2. Comparison based on the CEC 2014 suite

In this experiment, our algorithm is compared with the nine population-based metaheursitics mentioned above based on the CEC 2014 suite. The experimental results for the functions with 30, 50, and 100 in D are listed in Tables 3-5, respectively.

Table 3: Results of the ten algorithms for the CEC 2014 functions with 30 in dimensionality. "+" or "−" denotes that the current result is significantly better or statistical worse than the result of our algorithm in terms of Wilcoxon's rank sum test at a 0.05 significance level, respectively. Meanwhile, "≈" represents that there is no significant difference

Function	Average (standard deviation)									
	L-SHADE-EpSin	UMOEAs-II	MPEDE	ETI-JADE	HS-ES	AMECoDEs	EDEV	MLCC-SI	NDE	HS-ES-DE
F1	7.579E-15	1.960E-11	1.386E-04	5.819E+02	3.571E-10	1.137E-14	3.259E+03	2.982E+03	5.498E+02	3.174E-14
	(7.211E-15)+	(3.251E-11)−	(5.288E-04)≈	(1.075E+03)−	(3.199E-10)−	(5.782E-15)+	(9.097E+03)−	(3.293E+03)−	(1.314E+03)−	(2.033E-14)
F2	0.000E+00	1.137E-14	0.000E+00	0.000E+00	3.272E-10	1.895E-15	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	(0.000E+00)≈	(1.416E-14)−	(0.000E+00)≈	(0.000E+00)≈	(6.497E-10)−	(7.211E-15)≈	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)
F3	0.000E+00	7.579E-15	4.358E-14	4.351E-04	3.098E-10	7.579E-15	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	(0.000E+00)≈	(1.965E-14)−	(2.445E-14)−	(8.813E-04)−	(4.597E-10)−	(1.965E-14)−	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)
F4	2.653E-13	4.358E-14	2.201E-04	3.979E-14	3.049E-11	1.137E-14	6.863E-03	7.281E-07	7.738E-04	0.000E+00
	(1.004E-12)−	(3.231E-14)−	(1.099E-03)−	(3.702E-14)−	(4.406E-11)−	(2.313E-14)−	(2.553E-02)−	(3.384E-06)−	(1.532E-03)−	(0.000E+00)
F5	2.011E+01	2.000E+01	2.038E+01	2.001E+01	2.000E+01	2.000E+01	2.039E+01	2.021E+01	2.016E+01	2.000E+01
	(2.225E-02)−	(7.123E-04)+	(4.902E-02)−	(2.379E-02)−	(5.649E-04)≈	(1.135E-03)−	(5.110E-02)−	(5.611E-02)−	(8.502E-02)−	(6.995E-04)
F6	1.506E-05	2.288E-05	5.665E+00	5.446E-01	7.891E-01	5.432E-01	5.209E-01	1.097E+00	2.377E+00	8.279E-01
	(8.250E-05)+	(9.533E-05)≈	(4.667E+00)−	(6.927E-01)≈	(1.226E+00)≈	(1.001E+00)≈	(1.159E+00)≈	(1.837E+00)≈	(2.285E+00)−	(9.037E-01)
F7	0.000E+00	3.790E-15	3.286E-04	2.465E-04	1.972E-11	0.000E+00	0.000E+00	7.579E-15	0.000E+00	0.000E+00
	(0.000E+00)≈	(2.076E-14)≈	(1.800E-03)≈	(1.350E-03)≈	(6.250E-11)−	(0.000E+00)≈	(0.000E+00)≈	(2.884E-14)≈	(0.000E+00)≈	(0.000E+00)
F8	1.023E-13	9.095E-14	3.790E-15	0.000E+00	8.689E+00	0.000E+00	0.000E+00	1.895E-14	2.322E+00	3.283E+00
	(3.469E-14)+	(4.625E-14)+	(2.076E-14)+	(0.000E+00)+	(2.625E+00)−	(0.000E+00)+	(0.000E+00)+	(4.309E-14)+	(2.925E+00)+	(1.386E+00)
F9	1.319E+01	7.628E-01	2.696E+01	2.111E+01	7.064E+00	4.088E+01	3.483E+01	2.176E+01	4.503E+01	6.803E+00
	(2.321E+00)−	(8.540E-01)+	(7.257E+00)−	(4.897E+00)−	(2.618E+00)≈	(1.230E+01)−	(5.457E+00)−	(3.603E+00)−	(2.150E+01)−	(2.098E+00)
F10	2.082E-03	2.082E-03	3.083E+01	3.886E-02	4.174E+02	1.773E+00	2.144E+00	6.459E-01	4.374E+00	6.387E+01
	(6.353E-03)+	(8.381E-03)+	(1.176E+01)+	(2.366E-02)+	(2.330E+02)−	(4.040E-01)+	(9.488E+00)+	(6.092E-01)≈	(2.274E+01)+	(1.089E+02)
F11	1.171E+03	1.219E+03	3.040E+03	1.099E+03	6.849E+02	1.946E+03	2.509E+03	1.677E+03	2.101E+03	6.092E+02
	(1.922E+02)−	(2.345E+02)−	(4.000E+02)−	(3.750E+02)−	(3.326E+02)≈	(4.963E+02)−	(6.788E+02)−	(3.400E+02)−	(5.590E+02)−	(2.572E+02)
F12	1.639E-01	9.378E-02	5.227E-01	8.331E-02	1.693E-02	4.196E-02	5.534E-01	2.554E-01	1.553E-01	1.692E-02
	(1.771E-02)−	(5.496E-02)−	(9.219E-02)−	(2.843E-02)−	(1.506E-02)≈	(1.816E-02)−	(1.656E-01)−	(6.465E-02)−	(1.019E-01)−	(1.530E-02)
F13	1.325E-01	1.027E-01	2.110E-01	1.274E-01	4.905E-02	1.271E-01	1.935E-01	1.827E-01	9.899E-02	5.765E-02
	(2.147E-02)−	(2.671E-02)−	(3.586E-02)−	(3.223E-02)−	(1.172E-02)+	(2.515E-02)−	(3.356E-02)−	(2.387E-02)−	(3.031E-02)−	(1.630E-02)
F14	2.023E-01	2.342E-01	2.650E-01	1.700E-01	3.312E-01	2.216E-01	1.750E-01	2.058E-01	2.262E-01	2.173E-01
	(1.796E-02)≈	(2.409E-02)−	(3.039E-02)−	(2.209E-02)+	(4.970E-02)−	(2.802E-02)≈	(2.561E-02)+	(2.928E-02)≈	(3.656E-02)≈	(3.024E-02)
F15	2.317E+00	2.017E+00	4.028E+00	2.565E+00	3.014E+00	3.050E+00	4.029E+00	2.381E+00	3.098E+00	2.302E+00
	(2.404E-01)≈	(3.777E-01)+	(8.097E-01)−	(5.117E-01)−	(6.550E-01)−	(7.261E-01)−	(4.781E-01)−	(5.504E-01)≈	(8.706E-01)−	(3.029E-01)
F16	8.275E+00	9.024E+00	1.067E+01	8.195E+00	1.006E+01	1.020E+01	9.860E+00	9.419E+00	1.001E+01	8.186E+00
	(4.005E-01)≈	(6.421E-01)−	(4.629E-01)−	(7.204E-01)≈	(9.200E-01)−	(1.150E+00)−	(4.186E-01)−	(4.113E-01)−	(7.255E-01)−	(4.369E-01)
F17	1.450E+02	1.839E+02	2.559E+02	1.287E+04	2.132E+01	3.720E+02	4.516E+03	2.190E+02	1.911E+02	5.354E+02
	(7.455E+01)+	(9.060E+01)+	(1.890E+02)+	(4.573E+04)−	(3.919E+01)+	(1.824E+02)+	(8.345E+03)−	(1.141E+02)+	(1.188E+02)+	(2.847E+02)
F18	6.140E+00	5.044E+00	1.144E+01	1.738E+02	5.724E+00	1.924E+01	2.438E+01	9.581E+00	8.581E+00	6.824E+00
	(2.329E+00)≈	(2.491E+00)+	(5.214E+00)−	(6.441E+02)−	(2.761E+00)≈	(6.766E+00)−	(1.585E+01)−	(3.859E+00)−	(3.767E+00)−	(3.374E+00)
F19	2.457E+00	3.030E+00	9.150E+00	4.145E+00	2.941E+00	2.721E+00	3.804E+00	3.109E+00	2.824E+00	2.170E+00
	(6.729E-01)≈	(7.932E-01)−	(1.501E+00)−	(5.413E-01)−	(8.011E-01)−	(5.402E-01)−	(2.256E+00)−	(4.008E-01)−	(7.106E-01)−	(6.762E-01)
F20	2.304E+00	3.588E+00	7.385E+00	1.458E+02	3.646E+00	7.022E+00	1.474E+01	5.787E+00	5.298E+00	7.665E+00
	(9.133E-01)+	(1.259E+00)+	(1.903E+00)≈	(1.416E+02)−	(8.233E+00)+	(4.049E+00)≈	(3.599E+00)−	(2.114E+00)+	(1.697E+00)+	(3.330E+00)
F21	9.962E+01	5.469E+01	9.400E+01	7.733E+03	2.066E+01	1.400E+02	3.617E+02	1.015E+02	4.058E+01	1.779E+02
	(8.274E+01)+	(6.400E+01)+	(7.693E+01)+	(2.400E+04)−	(6.055E+01)+	(1.125E+02)≈	(1.713E+02)−	(7.229E+01)+	(5.136E+01)+	(9.995E+01)
F22	7.622E+01	2.758E+01	9.878E+01	1.085E+02	1.636E+02	1.960E+02	1.025E+02	6.862E+01	6.398E+01	1.442E+02
	(5.714E+01)+	(6.156E+00)+	(6.610E+01)≈	(7.497E+01)≈	(7.391E+01)≈	(1.306E+02)−	(6.141E+01)≈	(5.767E+01)+	(6.827E+01)+	(5.450E+00)
F23	2.000E+02	2.000E+02	3.140E+02	3.152E+02	3.152E+02	3.152E+02	3.140E+02	3.152E+02	3.152E+02	2.000E+02
	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)−	(5.782E-14)−	(7.707E-06)−	(5.782E-14)−	(1.573E-13)−	(5.782E-14)−	(1.455E-13)−	(0.000E+00)
F24	2.000E+02	2.000E+02	2.255E+02	2.249E+02	2.244E+02	2.236E+02	2.242E+02	2.230E+02	2.207E+02	2.000E+02
	(6.004E-10)≈	(8.444E-14)≈	(3.431E+00)−	(2.447E+00)−	(2.324E+00)−	(8.835E-01)−	(9.597E-01)−	(7.671E-01)−	(6.986E+00)−	(2.146E-10)
F25	2.000E+02	2.000E+02	2.002E+02	2.038E+02	2.086E+02	2.028E+02	2.002E+02	2.029E+02	2.028E+02	2.000E+02
	(0.000E+00)≈	(0.000E+00)≈	(2.430E-03)−	(1.074E+00)−	(1.921E+00)−	(2.914E-01)−	(2.738E-02)−	(2.666E-01)−	(2.876E-01)−	(0.000E+00)
F26	1.001E+02	1.001E+02	1.002E+02	1.001E+02	1.371E+02	1.001E+02	1.002E+02	1.002E+02	1.001E+02	1.434E+02
	(1.562E-02)+	(2.502E-02)+	(2.813E-02)≈	(3.439E-02)+	(4.315E+01)≈	(1.959E-02)+	(3.865E-02)≈	(2.599E-02)≈	(5.815E-02)+	(5.032E+01)
F27	2.033E+02	2.000E+02	4.865E+02	3.190E+02	3.021E+02	3.569E+02	3.600E+02	3.569E+02	4.004E+02	2.098E+02
	(1.826E+01)≈	(0.000E+00)+	(1.503E+02)−	(3.683E+01)−	(1.158E+01)−	(5.061E+01)−	(4.947E+01)−	(5.061E+01)−	(2.397E-01)−	(3.908E+01)
F28	2.000E+02	2.000E+02	3.704E+02	7.657E+02	8.972E+02	8.182E+02	3.825E+02	7.956E+02	8.260E+02	2.000E+02
	(0.000E+00)≈	(0.000E+00)≈	(2.056E+00)−	(6.146E+01)−	(2.503E+01)−	(2.605E+01)−	(6.518E+00)−	(1.740E+01)−	(2.681E+01)−	(0.000E+00)
F29	2.000E+02	7.161E+02	2.111E+02	7.177E+02	2.582E+02	7.162E+02	2.136E+02	6.734E+02	6.465E+02	2.000E+02
	(0.000E+00)≈	(2.564E+00)−	(3.118E+00)−	(3.883E+01)−	(7.806E+01)−	(2.220E+00)−	(9.651E-01)−	(1.756E+02)−	(1.843E+02)−	(0.000E+00)
F30	2.000E+02	9.258E+02	3.233E+02	1.811E+03	1.747E+03	9.599E+02	3.664E+02	4.827E+02	5.617E+02	2.000E+02
	(8.444E-14)≈	(3.513E+02)−	(6.869E+01)−	(7.022E+02)−	(3.984E+02)−	(3.380E+02)−	(1.070E+02)−	(1.042E+02)−	(1.856E+02)−	(6.581E-05)
−	6	12	20	21	18	19	21	17	19	
+	9	12	4	4	4	5	3	5	7	
≈	15	6	6	5	8	6	6	8	4	

Table 4: Results of the ten algorithms for the CEC 2014 functions with 50 in dimensionality. "+" or "−" denotes that the current result is significantly better or statistical worse than the result of our algorithm in terms of Wilcoxon's rank sum test at a 0.05 significance level, respectively. Meanwhile, "≈" represents that there is no significant difference

Function	Average (standard deviation)									
	L-SHADE-EpSin	UMOEAs-II	MPEDE	ETI-JADE	HS-ES	AMECoDEs	EDEV	MLCC-SI	NDE	HS-ES-DE
F1	9.125E-05	1.788E-03	8.062E+04	1.472E+04	2.258E-09	2.644E+04	2.148E+04	2.944E+05	1.066E+05	1.594E-12
	(2.613E-04)−	(1.051E-03)−	(5.965E+04)−	(1.267E+04)−	(3.469E-09)−	(1.710E+04)−	(1.879E+04)−	(1.000E+05)−	(4.911E+04)−	(1.266E-12)
F2	3.316E-14	2.624E-13	5.305E-14	9.853E-14	1.873E-09	3.221E-14	1.326E-14	2.271E-04	3.840E-03	4.453E-14
	(1.077E-14)≈	(1.936E-13)−	(1.624E-14)≈	(3.048E-13)≈	(2.328E-09)−	(9.827E-15)≈	(1.937E-14)+	(1.899E-04)−	(7.727E-03)−	(1.615E-14)
F3	5.684E-14	1.137E-13	8.633E-05	2.619E+02	1.200E-09	1.042E-13	6.126E-03	6.037E-01	5.497E-05	5.684E-14
	(0.000E+00)+	(3.657E-14)−	(1.377E-04)−	(2.025E+02)−	(1.750E-09)−	(5.987E-14)≈	(3.017E-02)−	(3.306E+00)−	(7.699E-05)−	(0.000E+00)
F4	5.313E+01	3.926E+01	9.037E+00	1.981E+01	3.254E+00	4.649E+01	4.929E-01	6.301E+01	5.525E+01	1.762E-13
	(4.893E+01)−	(4.891E+01)−	(3.080E+00)−	(3.983E+01)−	(1.783E+01)−	(4.742E+01)−	(1.790E+00)−	(3.114E+01)−	(4.171E+01)−	(1.016E-13)
F5	2.025E+01	2.000E+01	2.051E+01	2.000E+01	2.000E+01	2.001E+01	2.057E+01	2.032E+01	2.029E+01	2.000E+01
	(5.854E-02)−	(2.202E-04)+	(4.198E-02)−	(1.549E-03)−	(1.164E-04)≈	(1.094E-02)−	(1.041E-01)−	(7.044E-02)−	(1.208E-01)−	(1.292E-04)
F6	6.942E-02	1.606E-01	1.162E+01	1.988E+00	4.070E-05	1.591E+00	3.266E+00	1.039E-01	9.799E+00	2.599E-01
	(1.795E-01)+	(3.198E-01)+	(3.806E+00)−	(1.595E+00)−	(8.368E-05)+	(1.046E+00)−	(2.084E+00)−	(2.075E-01)+	(4.097E+00)−	(5.616E-01)
F7	1.023E-13	1.137E-13	1.808E-03	2.136E-03	9.665E-10	8.216E-04	2.547E-03	1.137E-13	1.068E-03	0.000E+00
	(3.469E-14)−	(0.000E+00)−	(3.715E-03)−	(4.028E-03)−	(2.188E-09)−	(2.535E-03)−	(4.597E-03)−	(0.000E+00)−	(2.798E-03)−	(0.000E+00)
F8	5.122E-10	2.107E-12	1.137E-13	7.579E-15	1.526E+00	9.474E-14	0.000E+00	2.361E-09	8.822E+00	1.254E+01
	(1.021E-09)+	(1.114E-12)+	(0.000E+00)+	(2.884E-14)+	(1.273E+00)+	(4.309E-14)+	(0.000E+00)+	(3.422E-09)+	(5.191E+00)+	(4.036E+00)
F9	2.709E+01	4.146E+00	5.323E+01	4.643E+01	8.291E-01	5.550E+01	6.201E+01	4.833E+01	5.850E+01	1.932E+01
	(7.139E+00)−	(1.457E+00)+	(1.041E+01)−	(8.765E+00)−	(7.429E-01)+	(1.516E+01)−	(1.154E+01)−	(8.437E+00)−	(3.194E+01)−	(4.587E+00)
F10	5.186E-02	5.356E-01	3.074E+01	8.286E-02	2.038E+02	1.391E+01	7.840E-02	7.965E+00	1.213E+01	4.045E+02
	(2.122E-02)+	(5.694E-01)+	(1.422E+01)+	(3.586E-02)+	(1.591E+02)≈	(1.941E+00)+	(9.343E-02)+	(2.945E+00)+	(6.579E+01)+	(3.440E+02)
F11	3.061E+03	3.503E+03	6.325E+03	2.705E+03	5.548E+02	3.733E+03	5.706E+03	3.978E+03	4.945E+03	1.399E+03
	(3.020E+02)−	(5.835E+02)−	(6.945E+02)−	(6.353E+02)−	(2.348E+02)+	(9.909E+02)−	(1.665E+03)−	(5.715E+02)−	(8.093E+02)−	(5.386E+02)
F12	2.126E-01	1.229E-01	6.446E-01	6.838E-02	2.477E-02	5.562E-02	8.023E-01	3.302E-01	1.461E-01	1.546E-02
	(2.664E-02)−	(7.328E-02)−	(1.064E-01)−	(2.844E-02)−	(2.376E-02)≈	(1.548E-02)−	(3.463E-01)−	(6.663E-02)−	(1.097E-01)−	(1.250E-02)
F13	2.088E-01	1.499E-01	2.900E-01	2.211E-01	7.023E-02	1.981E-01	3.057E-01	2.834E-01	1.602E-01	1.359E-01
	(1.641E-02)−	(3.260E-02)≈	(3.993E-02)−	(3.726E-02)−	(1.186E-02)+	(2.919E-02)−	(4.528E-02)−	(3.485E-02)−	(4.191E-02)−	(2.565E-02)
F14	1.942E-01	2.908E-01	2.960E-01	1.986E-01	3.988E-01	3.031E-01	2.512E-01	2.697E-01	2.831E-01	2.606E-01
	(2.056E-02)+	(2.743E-02)−	(3.476E-02)−	(3.156E-02)+	(4.127E-02)−	(3.263E-02)−	(2.944E-02)≈	(2.215E-02)≈	(4.224E-02)−	(3.116E-02)
F15	5.437E+00	5.274E+00	6.491E+00	5.004E+00	4.736E+00	5.922E+00	9.165E+00	6.614E+00	5.487E+00	4.611E+00
	(6.345E-01)−	(9.345E-01)−	(2.007E+00)−	(1.000E+00)−	(9.544E-01)≈	(1.050E+00)−	(1.135E+00)−	(1.379E+00)−	(1.041E+00)−	(4.973E-01)
F16	1.670E+01	1.804E+01	1.927E+01	1.629E+01	1.863E+01	1.876E+01	1.853E+01	1.842E+01	1.887E+01	1.621E+01
	(4.526E-01)−	(1.027E+00)−	(5.247E-01)−	(1.127E+00)≈	(8.725E-01)−	(1.071E+00)−	(6.409E-01)−	(4.335E-01)−	(6.354E-01)−	(5.559E-01)
F17	3.335E+02	1.108E+03	1.962E+03	2.396E+03	1.596E+03	2.036E+03	7.458E+03	1.160E+03	9.095E+02	1.603E+03
	(1.397E+02)+	(3.853E+02)+	(5.862E+02)−	(6.505E+02)−	(1.330E+03)≈	(3.949E+02)−	(1.588E+04)−	(3.665E+02)+	(3.767E+02)+	(3.961E+02)
F18	1.841E+01	7.297E+01	1.399E+02	1.822E+02	8.301E-01	1.247E+02	1.398E+02	3.684E+01	3.141E+01	2.190E+01
	(4.616E+00)+	(1.920E+01)−	(5.867E+01)−	(5.271E+01)−	(6.565E-01)+	(2.795E+01)−	(7.957E+01)−	(9.040E+00)−	(1.059E+01)−	(6.292E+00)
F19	9.908E+00	8.331E+00	1.161E+01	7.553E+00	7.145E+00	5.934E+00	2.269E+01	9.973E+00	1.006E+01	5.902E+00
	(6.301E-01)−	(2.190E+00)−	(3.087E+00)−	(2.748E+00)−	(9.264E-01)−	(6.735E-01)≈	(2.240E+01)−	(4.701E-01)−	(6.840E-01)−	(8.251E-01)
F20	5.664E+00	1.402E+01	6.228E+01	5.226E+02	2.267E+00	6.536E+01	5.903E+01	2.601E+01	3.219E+01	8.656E+01
	(1.469E+00)+	(4.798E+00)+	(3.807E+01)+	(3.607E+02)−	(5.798E-01)+	(2.404E+01)+	(4.815E+01)+	(6.282E+00)+	(1.138E+01)+	(3.128E+01)
F21	3.522E+02	4.819E+02	7.776E+02	1.903E+04	1.378E+03	8.932E+02	2.813E+03	4.735E+02	4.859E+02	1.013E+03
	(1.082E+02)+	(1.273E+02)+	(2.254E+02)+	(7.486E+04)−	(5.512E+02)−	(3.198E+02)≈	(5.240E+03)−	(1.315E+02)+	(1.579E+02)+	(2.690E+02)
F22	1.098E+02	1.512E+02	5.573E+02	4.618E+02	1.606E+02	5.710E+02	3.851E+02	2.822E+02	4.022E+02	1.704E+02
	(7.371E+01)≈	(8.327E+01)≈	(1.919E+02)−	(1.734E+02)−	(5.624E+01)≈	(2.376E+02)−	(1.607E+02)−	(1.220E+02)−	(2.490E+02)−	(7.905E+01)
F23	2.000E+02	2.000E+02	3.370E+02	3.440E+02	3.440E+02	3.440E+02	3.370E+02	3.440E+02	3.440E+02	2.000E+02
	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)−	(3.156E-13)−	(2.192E-05)−	(1.734E-13)−	(6.443E-13)−	(2.598E-13)−	(2.891E-13)−	(0.000E+00)
F24	2.000E+02	2.000E+02	2.774E+02	2.737E+02	2.686E+02	2.732E+02	2.728E+02	2.579E+02	2.708E+02	2.000E+02
	(3.929E-09)≈	(1.154E-13)≈	(1.453E+00)−	(2.669E+00)−	(1.721E+00)−	(1.712E+00)−	(2.935E+00)−	(1.884E+00)−	(2.561E+00)−	(2.117E-10)
F25	2.000E+02	2.000E+02	2.003E+02	2.163E+02	2.159E+02	2.070E+02	2.184E+02	2.065E+02	2.060E+02	2.000E+02
	(0.000E+00)≈	(0.000E+00)≈	(2.549E-02)−	(6.935E+00)−	(3.229E+00)−	(3.437E+00)−	(9.367E+00)−	(1.033E+00)−	(5.869E-01)−	(0.000E+00)
F26	1.002E+02	1.001E+02	1.003E+02	1.036E+02	1.183E+02	1.002E+02	1.402E+02	1.003E+02	1.002E+02	1.036E+02
	(1.996E-02)+	(3.587E-02)+	(2.991E-02)+	(1.821E+01)≈	(3.261E+01)−	(3.361E-02)+	(4.967E+01)−	(2.952E-02)≈	(8.257E-02)≈	(1.821E+01)
F27	2.033E+02	2.000E+02	8.679E+02	4.407E+02	3.000E+02	3.498E+02	4.177E+02	3.090E+02	3.841E+02	2.033E+02
	(1.826E+01)≈	(0.000E+00)≈	(2.358E+02)−	(5.567E+01)−	(2.167E-03)−	(3.182E+01)−	(9.408E+01)−	(1.825E+01)−	(5.297E+01)−	(1.826E+01)
F28	2.000E+02	2.000E+02	3.662E+02	1.089E+03	1.226E+03	1.106E+03	3.741E+02	1.144E+03	1.140E+03	2.000E+02
	(0.000E+00)≈	(0.000E+00)≈	(3.082E+00)−	(4.562E+01)−	(7.565E+01)−	(3.877E+01)−	(6.810E+00)−	(3.563E+01)−	(4.267E+01)−	(0.000E+00)
F29	2.000E+02	7.977E+02	2.077E+02	8.908E+02	4.890E+02	8.120E+02	2.253E+02	5.890E+02	7.630E+02	2.000E+02
	(0.000E+00)≈	(2.950E+01)−	(2.017E+00)−	(6.270E+01)−	(3.018E+01)−	(3.958E+01)−	(3.055E+00)−	(1.112E+02)−	(6.117E+01)−	(0.000E+00)
F30	2.000E+02	8.838E+03	8.742E+02	9.710E+03	8.768E+03	8.896E+03	9.310E+02	8.539E+03	8.999E+03	2.000E+02
	(0.000E+00)≈	(5.284E+02)−	(2.107E+02)−	(7.522E+02)−	(3.676E+02)−	(5.571E+02)−	(1.698E+02)−	(4.006E+02)−	(5.673E+02)−	(0.000E+00)
−	11	14	24	24	17	22	25	22	24	
+	10	9	5	3	7	4	4	6	5	
≈	9	7	1	3	6	4	1	2	1	

Table 5: Results of the ten algorithms for the CEC 2014 functions with 100 in dimensionality. "+" or "−" denotes that the current result is significantly better or statistical worse than the result of our algorithm in terms of Wilcoxon's rank sum test at a 0.05 significance level, respectively. Meanwhile, "≈" represents that there is no significant difference

Function	Average (standard deviation)									
	L-SHADE-EpSin	UMOEAs-II	MPEDE	ETI-JADE	HS-ES	AMECoDEs	EDEV	MLCC-SI	NDE	HS-ES-DE
F1	1.567E+04	3.977E-03	2.466E+05	1.252E+05	1.202E-01	1.959E+05	1.201E+05	5.818E+06	9.691E+05	1.209E-01
	(9.141E+03)−	(5.433E-04)+	(1.094E+05)−	(8.276E+04)−	(1.771E-01)≈	(6.112E+04)−	(6.952E+04)−	(1.077E+06)−	(2.990E+05)−	(9.925E-02)
F2	5.381E-13	1.828E-09	2.084E-13	4.180E-10	1.991E-09	2.002E-12	2.417E-10	1.343E+02	1.898E+03	2.554E-12
	(4.416E-13)≈	(3.213E-09)−	(1.193E-13)+	(1.226E-09)−	(3.114E-09)−	(1.015E-11)+	(4.096E-10)−	(7.603E+01)−	(2.767E+03)−	(4.839E-12)
F3	3.790E-13	8.522E-09	6.584E+02	1.763E+02	1.200E-08	3.070E-03	2.772E+00	1.590E+00	1.099E+01	1.016E-10
	(1.027E-13)+	(1.289E-08)−	(9.622E+02)−	(1.619E+02)−	(2.340E-08)−	(1.088E-02)−	(4.667E+00)−	(1.336E+00)−	(1.055E+01)−	(6.025E-11)
F4	1.765E+02	1.639E+02	6.245E+01	8.381E+01	5.392E+01	1.031E+02	2.106E+01	1.710E+02	1.726E+02	4.854E+01
	(3.059E+01)−	(3.071E+01)−	(3.144E+01)≈	(5.081E+01)−	(6.836E+01)≈	(4.127E+01)−	(4.615E+01)≈	(4.113E+01)−	(3.293E+01)−	(6.632E+01)
F5	2.056E+01	2.000E+01	2.078E+01	2.000E+01	2.000E+01	2.004E+01	2.078E+01	2.054E+01	2.048E+01	2.000E+01
	(4.743E-02)−	(1.070E-04)+	(4.224E-02)−	(4.158E-03)−	(1.036E-04)≈	(2.611E-02)−	(2.306E-01)−	(5.632E-02)−	(2.659E-01)−	(5.411E-05)
F6	3.567E-01	9.055E+00	5.495E+01	2.141E+01	1.637E+00	2.636E+01	3.690E+01	1.056E+01	5.788E+01	6.148E+00
	(4.910E-01)+	(2.671E+00)−	(7.274E+00)−	(9.510E+00)−	(9.162E-01)+	(2.999E+00)−	(4.760E+00)−	(2.435E+00)−	(1.598E+01)−	(2.007E+00)
F7	2.122E-13	3.070E-13	1.888E-03	5.751E-04	9.737E-09	1.151E-03	1.968E-03	7.511E-12	2.873E-03	1.781E-13
	(3.931E-14)≈	(9.512E-14)−	(4.866E-03)−	(2.212E-03)≈	(3.141E-08)−	(3.019E-03)−	(5.825E-03)−	(5.738E-12)−	(6.720E-03)−	(5.730E-14)
F8	3.526E-03	6.124E-12	2.322E-01	1.099E-13	3.880E+00	2.274E-13	0.000E+00	1.726E+01	2.867E+01	3.897E+01
	(2.388E-03)+	(2.183E-12)+	(5.015E-01)+	(2.076E-14)+	(1.149E+00)+	(0.000E+00)+	(0.000E+00)+	(4.067E+00)+	(2.506E+01)+	(8.188E+00)
F9	5.612E+01	2.560E+01	1.801E+02	1.427E+02	2.653E+00	9.764E+01	1.723E+02	1.249E+02	9.873E+01	7.058E+01
	(1.097E+01)+	(4.923E+00)+	(2.890E+01)−	(1.470E+01)−	(1.659E+00)+	(1.326E+01)−	(3.085E+01)−	(2.163E+01)−	(3.002E+01)−	(1.463E+01)
F10	1.968E+01	8.529E+01	5.066E+00	8.450E-01	6.192E+02	4.123E+01	7.874E-01	2.497E+02	3.927E+01	1.669E+03
	(4.315E+00)+	(1.359E+02)+	(2.553E+00)+	(8.016E-01)+	(3.222E+02)+	(3.593E+00)+	(2.636E-01)+	(1.160E+02)+	(2.290E+01)+	(4.281E+02)
F11	1.053E+04	1.162E+04	1.229E+04	9.304E+03	2.066E+03	1.076E+04	1.063E+04	1.308E+04	1.221E+04	6.167E+03
	(5.208E+02)−	(8.615E+02)−	(1.464E+03)−	(1.172E+03)−	(3.655E+02)+	(1.626E+03)−	(1.050E+03)−	(1.853E+03)−	(1.328E+03)−	(1.113E+03)
F12	4.361E-01	2.515E-01	1.092E+00	8.752E-02	1.662E-02	1.514E-01	8.641E-01	7.339E-01	2.882E-01	1.322E-02
	(4.035E-02)−	(1.143E-01)−	(6.026E-01)−	(3.132E-02)−	(1.553E-02)≈	(3.683E-02)−	(6.123E-01)−	(9.634E-02)−	(2.023E-01)−	(1.044E-02)
F13	3.225E-01	1.963E-01	3.942E-01	3.159E-01	8.783E-02	2.923E-01	3.838E-01	3.517E-01	2.860E-01	2.384E-01
	(2.239E-02)−	(4.381E-02)+	(4.715E-02)−	(3.328E-02)−	(1.547E-02)+	(3.354E-02)−	(3.941E-02)−	(3.270E-02)−	(5.836E-02)−	(3.486E-02)
F14	1.704E-01	3.251E-01	3.180E-01	2.346E-01	3.713E-01	3.144E-01	2.758E-01	2.960E-01	2.323E-01	2.680E-01
	(1.113E-02)+	(1.767E-02)−	(3.466E-02)−	(2.418E-02)+	(3.025E-02)−	(2.394E-02)−	(2.507E-02)≈	(1.636E-02)−	(2.342E-02)+	(1.764E-02)
F15	1.640E+01	1.256E+01	1.726E+01	1.960E+01	1.193E+01	1.478E+01	2.097E+01	3.114E+01	1.191E+01	1.124E+01
	(1.122E+00)−	(1.143E+00)−	(2.798E+00)−	(3.251E+00)−	(1.813E+00)≈	(4.561E+00)−	(3.128E+00)−	(5.603E+00)−	(1.510E+00)≈	(1.432E+00)
F16	3.877E+01	4.166E+01	4.102E+01	3.919E+01	4.115E+01	4.082E+01	4.021E+01	4.163E+01	4.248E+01	3.780E+01
	(6.385E-01)−	(1.188E+00)−	(6.775E-01)−	(2.085E+00)−	(1.128E+00)−	(1.829E+00)−	(7.755E-01)−	(7.186E-01)−	(1.183E+00)−	(9.214E-01)
F17	2.075E+03	4.314E+03	3.794E+04	1.306E+04	1.527E+03	1.651E+04	8.652E+03	4.283E+04	9.248E+03	4.865E+03
	(4.706E+02)+	(6.258E+02)+	(2.599E+04)−	(9.489E+03)−	(2.163E+03)+	(5.873E+03)−	(2.869E+03)−	(1.888E+04)−	(4.146E+03)−	(7.468E+02)
F18	8.995E+01	2.231E+02	4.509E+02	5.979E+02	1.659E+00	4.150E+02	3.577E+02	2.729E+02	2.530E+02	1.874E+02
	(1.846E+01)+	(1.078E+01)−	(2.925E+02)−	(4.762E+02)−	(9.665E-01)+	(2.075E+02)−	(1.125E+02)−	(1.644E+02)−	(2.943E+01)−	(4.158E+01)
F19	8.887E+01	9.674E+01	2.465E+01	9.526E+01	7.127E+01	9.625E+01	2.969E+01	9.248E+01	9.158E+01	7.332E+01
	(1.448E+00)−	(2.347E+00)−	(4.867E+00)+	(2.275E+01)−	(2.287E+01)≈	(4.854E+00)−	(1.224E+01)+	(2.764E+00)−	(1.570E+00)−	(2.361E+01)
F20	2.064E+01	1.385E+02	6.066E+02	1.688E+03	3.579E+02	5.433E+02	3.178E+02	2.057E+02	3.384E+02	3.408E+02
	(4.247E+00)+	(4.728E+01)+	(2.640E+02)−	(1.848E+03)−	(2.102E+02)−	(1.362E+02)−	(1.644E+02)≈	(4.135E+01)+	(9.327E+01)≈	(7.599E+01)
F21	5.848E+02	1.900E+03	6.911E+03	3.855E+03	3.199E+03	3.851E+03	5.375E+03	4.312E+03	1.723E+03	2.744E+03
	(2.342E+02)+	(5.520E+02)+	(5.869E+03)−	(1.464E+03)−	(6.708E+02)−	(1.095E+03)−	(8.583E+03)−	(1.570E+03)−	(4.510E+02)+	(5.709E+02)
F22	1.078E+03	1.048E+03	2.035E+03	1.399E+03	6.487E+02	1.373E+03	1.341E+03	1.243E+03	1.512E+03	2.991E+02
	(1.831E+02)−	(2.899E+02)−	(4.194E+02)−	(3.246E+02)−	(4.000E+02)−	(4.612E+02)−	(3.047E+02)−	(2.224E+02)−	(4.526E+02)−	(1.586E+02)
F23	2.000E+02	2.000E+02	3.450E+02	3.482E+02	3.483E+02	3.482E+02	3.450E+02	3.482E+02	3.482E+02	2.000E+02
	(0.000E+00)≈	(0.000E+00)≈	(4.264E-13)−	(8.444E-14)−	(3.827E-03)−	(2.828E-13)−	(4.679E-12)−	(3.947E-08)−	(3.839E-12)−	(0.000E+00)
F24	2.000E+02	2.000E+02	3.978E+02	3.993E+02	3.754E+02	3.891E+02	3.907E+02	3.568E+02	3.788E+02	2.000E+02
	(4.825E-11)≈	(1.059E-11)≈	(4.726E+00)−	(6.032E+00)−	(2.719E+00)−	(3.453E+00)−	(5.090E+00)−	(1.075E+00)−	(2.791E+00)−	(3.019E-10)
F25	2.000E+02	2.000E+02	2.201E+02	2.716E+02	2.014E+02	2.410E+02	2.344E+02	2.000E+02	2.261E+02	2.000E+02
	(0.000E+00)≈	(0.000E+00)≈	(2.352E+01)−	(6.217E+00)−	(2.903E+00)−	(1.709E+01)−	(2.381E+01)−	(2.850E-04)≈	(1.036E+01)−	(0.000E+00)
F26	2.000E+02	2.000E+02	1.536E+02	2.001E+02	2.000E+02	2.001E+02	2.001E+02	2.001E+02	1.568E+02	2.000E+02
	(0.000E+00)≈	(0.000E+00)≈	(5.055E+01)≈	(4.545E-03)−	(3.864E-02)−	(3.599E-02)−	(2.368E-02)−	(1.515E-02)−	(5.028E+01)≈	(0.000E+00)
F27	2.067E+02	2.000E+02	1.964E+03	1.031E+03	3.000E+02	6.903E+02	1.043E+03	3.659E+02	6.705E+02	2.033E+02
	(2.542E+01)≈	(0.000E+00)≈	(2.579E+02)−	(1.166E+02)−	(1.855E-05)−	(7.429E+01)−	(8.572E+01)−	(2.237E+01)−	(7.572E+01)−	(1.826E+01)
F28	2.000E+02	2.000E+02	5.253E+02	2.192E+03	2.347E+03	2.175E+03	6.546E+02	2.249E+03	2.273E+03	2.000E+02
	(0.000E+00)≈	(0.000E+00)≈	(5.214E+01)−	(1.824E+02)−	(2.696E+02)−	(1.084E+02)−	(8.904E+01)−	(5.373E+01)−	(5.619E+01)−	(0.000E+00)
F29	2.000E+02	8.701E+02	2.497E+02	1.311E+03	7.304E+02	9.949E+02	2.501E+02	1.863E+03	1.098E+03	2.000E+02
	(0.000E+00)≈	(1.093E+02)−	(5.733E+00)−	(1.732E+02)−	(5.883E+01)−	(2.210E+02)−	(4.787E+00)−	(1.142E+02)−	(1.861E+02)−	(0.000E+00)
F30	2.000E+02	7.927E+03	2.517E+03	8.681E+03	6.138E+03	7.738E+03	2.585E+03	6.309E+03	3.796E+03	2.000E+02
	(0.000E+00)≈	(9.336E+02)−	(4.646E+02)−	(1.435E+03)−	(1.562E+03)−	(1.121E+03)−	(5.129E+02)−	(5.514E+02)−	(7.949E+02)−	(0.000E+00)
−	10	15	24	26	16	27	24	26	23	
+	10	9	4	3	8	3	3	3	4	
≈	10	6	2	1	6	0	3	1	3	

According to Table 3 for the functions with 30 in dimensionality, our HS-ES-DE performs significantly better than L-SHADE-EpSin, UMOEAs-II, MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, and NDE on six, 12, 20, 21, 18, 19, 21, 17, and 19 functions, respectively. Meanwhile, HS-ES-DE is inferior to the peers on nine, 12, four, four, four, five, three, five, and seven functions, respectively. That is, although our algorithm loses to L-SHADE-EpSin and cannot defeat UMOEAs-II, it has better performance than the other all peers. More details are given below.

- For the three unimodal functions, HS-ES-DE is superior to UMOEAs-II on three functions (F1-F3), MPEDE on one function (F3), ETI-JADE on two functions (F1 and F3), HS-ES on three functions (F1-F3), AMECoDEs on one function (F3), EDEV on one function (F1), MLCC-SI on one function (F1), and NDE on one function (F1). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on one function (F1) and AMECoDEs on one function (F1). That is, our algorithm performs better than UMOEAs-II, MPEDE, ETI-JADE, HS-ES, EDEV, MLCC-SI, and NDE, but performs worse than L-SHADE-EpSin. Meanwhile, our algorithm is comparable to AMECoDEs.

- For the 13 simple multimodal functions, HS-ES-DE is superior to L-SHADE-EpSin on six functions (F4, F5, F9, and F11-F13), UMOEAs-II on six functions (F4, F11-F14, and F16), MPEDE on ten functions (F4-F6, F9, and F11-F16), ETI-JADE on seven functions (F4, F5, F9, F11-F13, and F15), HS-ES on seven functions (F4, F7-F8, F10, and F14-F16), AMECoDEs on eight functions (F4, F5, F9, F11-F13, F15 and F16), EDEV on eight functions (F4, F5, F9, F11-F13, F15, and F16), MLCC-SI on seven functions (F4, F5, F9, F11-F13, and F16), and NDE on nine functions (F4-F6, F9, F11-F13, F15, and F16). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on three functions (F6, F8, and F10), UMOEAs-II on five functions (F5, F8-F10, and F15), MPEDE on two functions (F8 and F10), ETI-JADE on three functions (F8, F10, and F14), HS-ES on one function (F13), AMECoDEs on two functions (F8 and F10), EDEV on three functions (F8, F10, and F14), MLCC-SI on one function (F8), and NDE on two functions (F8 and F10). That is, our algorithm performs better than all the peers.

- For the six hybrid functions, HS-ES-DE is superior to UMOEAs-II on one function (F19), MPEDE on two functions (F18 and F19), ETI-JADE on five functions (F17-F21), HS-ES

on one function (F19), AMECoDEs on three functions (F18, F19, and F22), EDEV on five functions (F17-F21), MLCC-SI on two functions (F18 and F19), and NDE two functions (F18 and F19). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on four functions (F17, and F20-F22), UMOEAs-II on five functions (F17, F18, and F20-F22), MPEDE on two functions (F17 and F22), HS-ES on three functions (F17, F20 and F21), AMECoDEs on one function (F17), MLCC-SI on four functions (F17 and F20-F22), and NDE on four functions (F17 and F20-F22). That is, our algorithm performs better than ETI-JADE, and AMECoDEs, EDEV, but performs worse than L-SHADE-EpSin, UMOEAs-II, HS-ES, MLCC-SI, and NDE. Meanwhile, our algorithm is comparable to MPEDE.

- For the eight composition functions, HS-ES-DE is superior to UMOEAs-II on two functions (F29 and F30), MPEDE on seven functions (F23-F25, and F27-F30), ETI-JADE on seven functions (F23-F25, and F27-F30), HS-ES on seven functions (F23-F25, and F27-F30), AMECoDEs on on seven functions (F23-F25, and F27-F30), EDEV on seven functions (F23-F25, and F27-F30), MLCC-SI on seven functions (F23-F25, and F27-F30), and NDE on seven functions (F23-F25, and F27-F30). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on one function (F26), UMOEAs-II on two functions (F26 and F27), ETI-JADE on one function (F26), AMECoDEs on one function (F26), and NDE on one function (26). That is, our algorithm performs better than MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, and NDE, but performs worse than L-SHADE-EpSin. Meanwhile, our algorithm is comparable to UMOEAs-II.

According to Table 4 for the functions with 50 in dimensionality, our HS-ES-DE performs significantly better than L-SHADE-EpSin, UMOEAs-II, MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, and NDE on 11, 14, 24, 24, 17, 22, 25, 22, and 24 functions, respectively. Meanwhile, HS-ES-DE is inferior to the peers on ten, nine, five, three, seven, four, four, six, and five functions, respectively. That is, our algorithm has better performance than all of the other peers. More details are given below.

- For the three unimodal functions, HS-ES-DE is superior to L-SHADE-EpSin on one function (F1), UMOEAs-II on three functions (F1-F3), MPEDE on two functions (F1 and F3), ETI-JADE on two functions (F1 and F3), HS-ES on three functions (F1-F3), AMECoDEs on one function (F1), EDEV on two functions (F1 and F3), MLCC-SI on three functions (F1-F3),

and NDE on three functions (F1-F3). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on one function (F3) and EDEV on one function (F2). That is, our algorithm performs better than UMOEAs-II, MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, NDE. Meanwhile, our algorithm is comparable to L-SHADE-EpSin.

- For the 13 simple multimodal functions, HS-ES-DE is superior to L-SHADE-EpSin on nine functions (F4, F5, F7, F9, F11-F13, F15 and F16), UMOEAs-II on seven functions (F4, F7, F11, F12, and F14-F16), MPEDE on 11 functions (F4-F7, F9, and F11-F16), ETI-JADE on nine functions (F4-F7, F9, F11-F13, and F15), HS-ES on four functions (F4, F7, F14, and F16), AMECoDEs on 11 functions (F4-F7, F9, and F11-F16), EDEV on ten functions (F4-F7, F9, F11-F13, F15, and F16), MLCC-SI on nine functions (F4, F5, F7, F9, F11-F13, F15, and F16), and NDE on 11 functions (F4-F7, F9, and F11-F16). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on four functions (F6, F8, F10, and F14), UMOEAs-II on five functions (F5, F6, and F8-F10), MPEDE on two functions (F8 and F10), ETI-JADE on three functions (F8, F10, and F14), HS-ES on five function (F6, F8, F9, F11, and F13), AMECoDEs on two functions (F8 and F10), EDEV on two functions (F8 and F10), MLCC-SI on three functions (F6, F8, and F10), and NDE on two functions (F8 and F10). That is, our algorithm performs better than L-SHADE-EpSin, UMOEAs-II, MPEDE, ETI-JADE, AMECoDEs, EDEV, MLCC-SI, NDE, but performs worse than HS-ES.

- For the six hybrid functions, HS-ES-DE is superior to L-SHADE-EpSin on one function (F19), UMOEAs-II on two functions (F18 and F19), MPEDE on four functions (F17-F19 and F22), ETI-JADE on six functions (F17-F22), HS-ES on two functions (F19 and F21), AMECoDEs on three functions (F17, F18, and F22), EDEV on five functions (F17-F19, F21, and F22), MLCC-SI on three functions (F18, F19, and F22), and NDE three functions (F18, F19, and F22). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on four functions (F17, F18, F20, and F21), UMOEAs-II on three functions (F17, F20, and F21), MPEDE on two functions (F20 and F21), HS-ES on two functions (F18 and F20), AMECoDEs on one function (F20), EDEV on one function (F20), MLCC-SI on two functions (F20 and F21), and NDE on three functions (F17, F20, and F21). That is, our algorithm performs better than MPEDE, ETI-JADE, AMECoDEs and EDEV, but performs worse than L-SHADE-EpSin, UMOEAs-II. Meanwhile, our algorithm is comparable to HS-ES, MLCC-SI, and NDE.

- For the eight composition functions, HS-ES-DE is superior to UMOEAs-II on two functions (F29 and F30), MPEDE on seven functions (F23-F25, and F27-F30), ETI-JADE on seven functions (F23-F25, and F27-F30), HS-ES on eight functions (F23-30), AMECoDEs on on seven functions (F23-F25, and F27-F30), EDEV on eight functions (F23-F30), MLCC-SI on seven functions (F23-F25, and F27-F30), and NDE on seven functions (F23-F25, and F27-F30). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on one function (F26), UMOEAs-II on one function (F26), MPEDE on one function (F26), and AMECoDEs on one function (F26). That is, our algorithm performs better than UMOEAs-II, MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, and NDE, but performs worse than L-SHADE-EpSin.

According to Table 5 for the functions with 100 in dimensionality, our HS-ES-DE performs significantly better than L-SHADE-EpSin, UMOEAs-II, MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, and NDE on ten, 15, 24, 26, 16, 27, 24, 26, and 23 functions, respectively. Meanwhile, HS-ES-DE is inferior to the peers on ten, nine, four, three, eight, three, three, three, and four functions, respectively. That is, although our algorithm cannot defeat L-SHADE-EpSin, it has better performance than all of the other peers. More details are given below.

- For the three unimodal functions, HS-ES-DE is superior to L-SHADE-EpSin on one function (F1), UMOEAs-II on two functions (F2 and F3), MPEDE on three functions (F1 and F3), ETI-JADE on three functions (F1-F3), HS-ES on two functions (F2 and F3), AMECoDEs on two functions (F1 and F3), EDEV on three functions (F1-F3), MLCC-SI on three functions (F1-F3), and NDE on three functions (F1-F3). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on one function (F3), UMOEAs-II on one function (F1), MPEDE on one function (F2), and AMECoDEs on one function (F2). That is, our algorithm performs better than UMOEAs-II, MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, and NDE. Meanwhile, our algorithm is comparable to L-SHADE-EpSin.

- For the 13 simple multimodal functions, HS-ES-DE is superior to L-SHADE-EpSin on seven functions (F4, F5, F11-F13, F15, and F16), UMOEAs-II on eight functions (F4, F6, F7, F11, F12, and F14-F16), MPEDE on ten functions (F5-F7, F9, and F11-F16), ETI-JADE on nine functions (F4-F6, F9, F11-F13, F15, and F16), HS-ES on three functions (F7, F14, and F16), AMECoDEs on 11 functions (F4-F7, F9, and F11-F16), EDEV on nine functions (F5-F7,

F9, F11-F13, F15, and F16), MLCC-SI on 11 functions (F4-F7, F9, and F11-F16), and NDE on nine functions (F4-F7, F9, F11-F13, and F16). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on five functions (F6, F8-F10 and F14), UMOEAs-II on five functions (F5, F8-F10, and F13), MPEDE on two functions (F8, and F10), ETI-JADE on three functions (F8, F10, and F14), HS-ES on six functions (F6, F8-F11, and F13), AMECoDEs on two functions (F8 and F10), EDEV on two functions (F8, and F10), MLCC-SI on two functions (F8 and F10), and NDE on three functions (F8, F10, and F14). That is, our algorithm performs better than L-SHADE-EpSin, UMOEAs-II, MPEDE, ETI-JADE, AMECoDEs, EDEV, MLCC-SI, and NDE, but performs worse than HS-ES.

- For the six hybrid functions, HS-ES-DE is superior to L-SHADE-EpSin on two functions (F19 and F22), UMOEAs-II on three functions (F18, F19, and F22), MPEDE on five functions (F17, F18, and F20-F22), ETI-JADE on six functions (F17-F22), HS-ES on three functions (F20-F22), AMECoDEs on six functions (F17-F22), EDEV on four functions (F17, F18, F21, and F22), MLCC-SI on five functions (F17-F19, F21, and F22), and NDE four functions (F17-F19 and F22). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on four functions (F17, F18, F20, and F21), UMOEAs-II on three functions (F17, F20, and F21), MPEDE on one function (F19), HS-ES on two functions (F17 and F18), EDEV on one function (F19), MLCC-SI on one function (F20), and NDE on one function (F21). That is, our algorithm performs better than MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI and NDE, but performs worse than L-SHADE-EpSin. Meanwhile, our algorithm is comparable to UMOEAs-II.

- For the eight composition functions, HS-ES-DE is superior to UMOEAs-II on two functions (F29 and F30), MPEDE on seven functions (F23-F25, and F27-F30), ETI-JADE on eight functions (F23-F30), HS-ES on eight functions (F23-F30), AMECoDEs on on eight functions (F23-F30), EDEV on eight functions (F23-F30), MLCC-SI on seven functions (F23, F24, and F26-F30), and NDE on seven functions (F23-F25, and F27-F30). That is, our algorithm performs better than UMOEAs-II, MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, and NDE. Meanwhile, our algorithm is comparable to L-SHADE-EpSin.

The Friedman test of this experiment is given in Table 6. When $D = 30$, our HS-ES-DE ranks third. When $D = 50$, our HS-ES-DE ranks second. Furthermore, HS-ES-DE ranks first when

Table 6: The Friedman test for the first experiment based on the CEC 2014 benchmark test suite

Algorithm	Ranking			Algorithm	Ranking		
	D=30	D=50	D=100		D=30	D=50	D=100
L-SHADE-EpSin	3.183	3.067	3.333	AMECoDEs	5.967	6.250	6.350
UMOEAs-II	3.633	3.867	4.267	EDEV	6.617	6.917	5.983
MPEDE	7.217	7.150	6.917	MLCC-SI	5.967	6.217	7.217
ETI-JADE	6.417	6.550	6.833	NDE	6.017	6.783	6.583
HS-ES	6.117	4.833	4.567	HS-ES-DE	3.867	3.366	3.100

$D = 100$.

Now that HS-ES-DE performs best when $D = 100$, we give convergence graphs when $D = 100$. In Figure 1, the convergence graph of the ten DE algorithms for 14 functions is given by plotting average of the best fitness in the 30 executions at intervals. For these functions, the average solution of our algorithm is not significantly worse than that of anyone among the peers.

It can be seen from Figure 1 that, in convergence graph of some algorithms, such as UMOEAs-II and HS-ES, the best fitness may go worse during the course of execution. The cause is that algorithm based on CMA-ES may not keep the ever best solution in execution. The phenomenon also exists in the earlier stage of convergence graph of our HS-ES-DE. After all, our algorithm begins with HS-ES. On the other hand, it can be seen that L-SHADE-EpSin can improve solution even in the final stage. So does our algorithm. For example, in Subfigures (g)-(i) and (k)-(n), the convergence graph of the two algorithms shows a suddenly decrease in the terminal part. After all, our algorithm ends with L-SHADE-EpSin.

4.3. Comparison based on the CEC 2017 suite

In this experiment, our algorithm is still compared with the nine population-based metaheuristics. The experimental results for the CEC 2017 functions with 30 in D are listed in Tables 7.

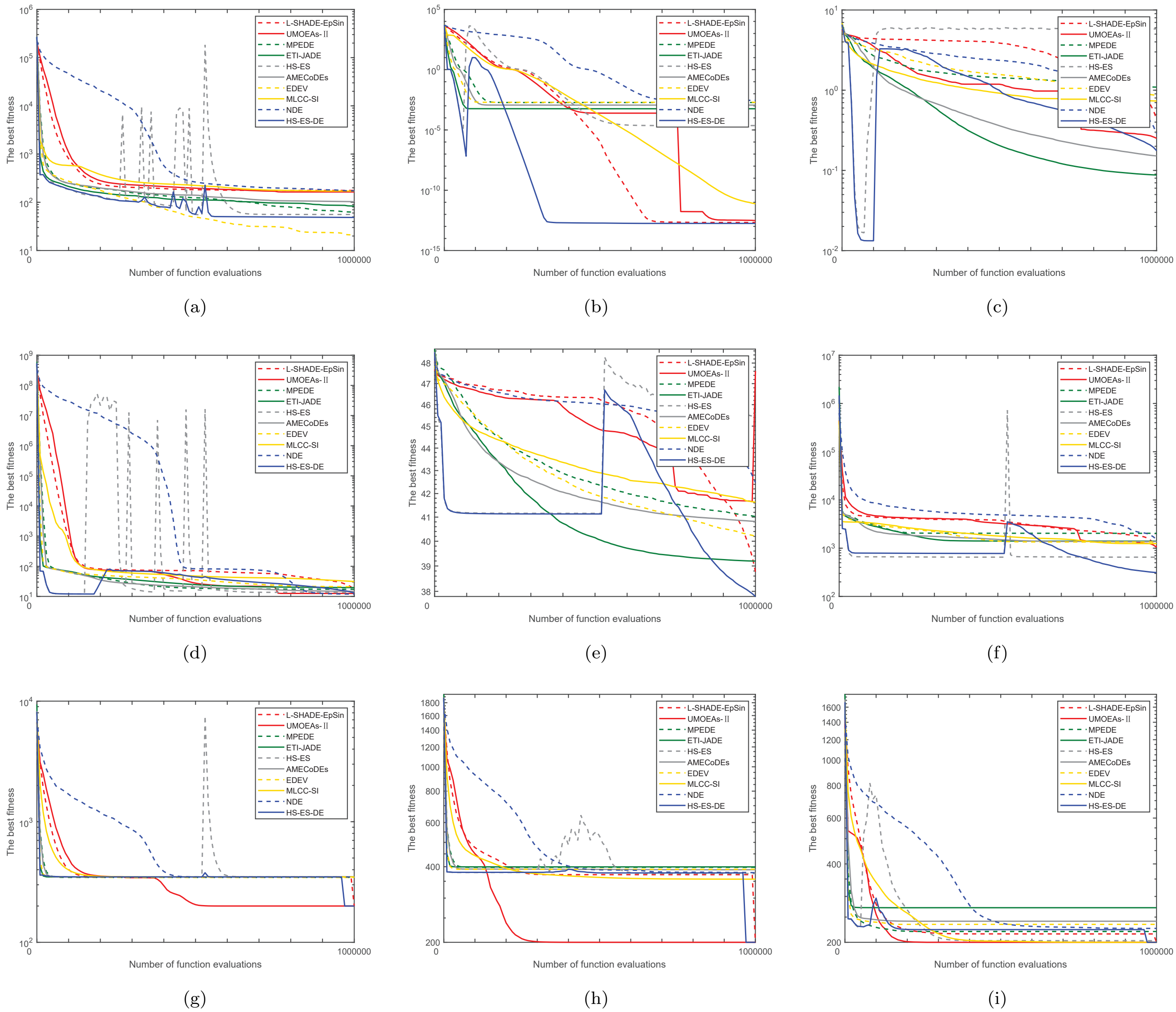
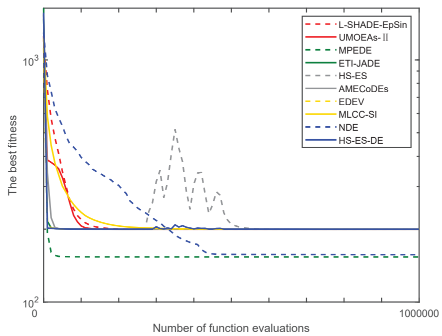
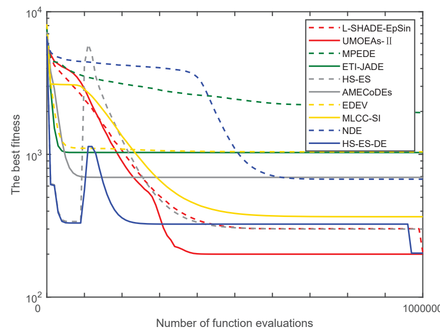


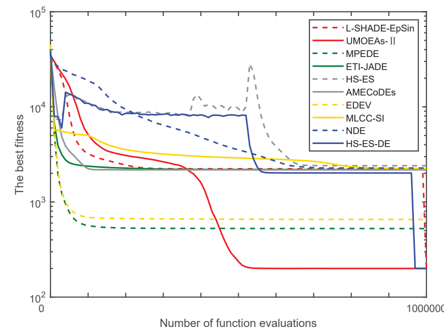
Figure 1: Convergence graphs of the ten algorithms for the 14 functions in the CEC 2014 benchmark test suite. (a): F4, (b): F7, (c): F12, (d): F15, (e): F16, (f): F22, (g): F23, (h): F24, (i): F25



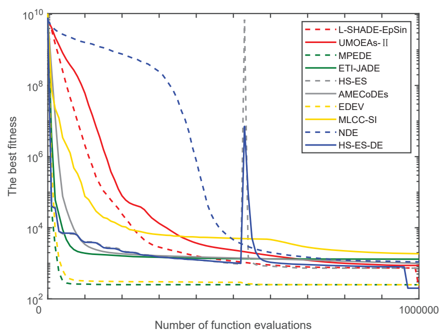
(j)



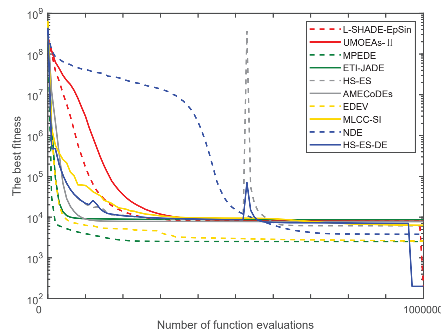
(k)



(l)



(m)



(n)

Figure 1: Convergence graphs of the ten algorithms for the eight functions in the CEC 2014 benchmark test suite. (j): F26, (k): F27, (l): F28, (m): F29, (n): F30

Table 7: Results of the ten algorithms for the CEC 2017 functions with 30 in dimensionality. ”+” or ”−” denotes that the current result is significantly better or statistical worse than the result of our algorithm in terms of Wilcoxon’s rank sum test at a 0.05 significance level, respectively. Meanwhile, ”≈” represents that there is no significant difference

Function	Average (standard deviation)									
	L-SHADE-EpSin	UMOEAs-II	MPEDE	ETI-JADE	HS-ES	AMECoDEs	EDEV	MLCC-SI	NDE	HS-ES-DE
F1	0.000E+00	9.000E-15	1.421E-15	3.316E-15	3.289E-10	1.421E-15	4.737E-16	5.211E-15	0.000E+00	4.263E-15
	(0.000E+00)+	(6.965E-15)−	(4.336E-15)≈	(6.113E-15)≈	(7.738E-10)−	(4.336E-15)≈	(2.594E-15)+	(6.965E-15)≈	(0.000E+00)+	(6.624E-15)
F2	1.895E-15	3.505E-14	1.042E-14	1.162E-12	1.260E-07	1.042E-14	5.161E+02	7.200E-13	4.838E-09	3.790E-15
	(7.211E-15)≈	(3.476E-14)−	(1.580E-14)−	(3.055E-12)−	(2.112E-07)−	(1.580E-14)−	(2.827E+03)−	(2.020E-12)−	(2.459E-08)−	(9.827E-15)
F3	7.579E-15	4.926E-14	8.602E-13	3.573E+03	3.648E-10	3.032E-14	1.637E-02	6.253E-14	6.184E-04	3.790E-14
	(1.965E-14)+	(2.468E-14)≈	(3.690E-12)−	(5.293E+03)≈	(6.235E-10)−	(2.884E-14)+	(6.369E-02)≈	(4.314E-14)−	(3.387E-03)−	(2.725E-14)
F4	5.875E+01	5.661E+01	6.644E-01	4.455E+01	3.056E+00	5.679E+01	1.728E+00	2.866E+01	5.912E+01	3.056E+00
	(1.014E+00)−	(1.069E+01)−	(1.511E+00)+	(2.745E+01)−	(1.715E+00)+	(1.077E+01)−	(2.009E+00)+	(2.927E+01)≈	(1.695E+00)−	(1.715E+00)
F5	1.249E+01	1.227E+00	2.285E+01	2.246E+01	8.557E+00	3.667E+01	3.196E+01	2.106E+01	3.976E+01	7.578E+00
	(1.742E+00)−	(1.129E+00)+	(5.083E+00)−	(5.630E+00)−	(3.109E+00)≈	(1.023E+01)−	(5.952E+00)−	(2.946E+00)−	(1.983E+01)−	(2.306E+00)
F6	4.604E-09	2.738E-08	1.027E-08	1.137E-13	1.137E-13	9.445E-07	1.137E-13	1.137E-13	4.563E-09	3.735E-09
	(2.498E-08)−	(6.749E-08)−	(3.499E-08)−	(0.000E+00)+	(0.000E+00)+	(3.069E-06)−	(0.000E+00)+	(0.000E+00)+	(2.499E-08)−	(1.107E-08)
F7	4.329E+01	3.296E+01	5.748E+01	5.070E+01	4.062E+01	6.227E+01	6.153E+01	5.272E+01	5.965E+01	3.982E+01
	(2.669E+00)−	(6.058E-01)+	(5.747E+00)−	(4.648E+00)−	(4.117E+00)≈	(7.295E+00)−	(5.759E+00)−	(4.288E+00)−	(1.031E+01)−	(3.198E+00)
F8	1.310E+01	1.194E+00	2.590E+01	2.259E+01	7.794E+00	3.887E+01	3.532E+01	2.286E+01	5.247E+01	6.926E+00
	(1.915E+00)−	(1.264E+00)+	(7.633E+00)−	(4.491E+00)−	(2.863E+00)≈	(1.029E+01)−	(5.630E+00)−	(3.327E+00)−	(1.911E+01)−	(2.089E+00)
F9	0.000E+00	0.000E+00	3.029E-02	1.514E-02	0.000E+00	0.000E+00	2.984E-03	0.000E+00	0.000E+00	0.000E+00
	(0.000E+00)≈	(0.000E+00)≈	(1.153E-01)≈	(8.295E-02)≈	(0.000E+00)≈	(0.000E+00)≈	(1.635E-02)≈	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)
F10	1.375E+03	1.468E+03	3.216E+03	1.247E+03	1.038E+03	2.249E+03	3.119E+03	1.973E+03	2.546E+03	5.939E+02
	(1.814E+02)−	(2.632E+02)−	(4.421E+02)−	(3.007E+02)−	(4.117E+02)−	(6.128E+02)−	(6.835E+02)−	(3.518E+02)−	(6.487E+02)−	(2.010E+02)
F11	9.728E+00	2.739E+01	1.459E+01	2.814E+01	8.120E+00	1.612E+01	2.852E+01	1.269E+01	1.349E+01	1.927E+01
	(1.740E+01)+	(2.878E+01)≈	(5.592E+00)+	(2.541E+01)−	(1.543E+01)+	(1.877E+01)+	(2.588E+01)−	(1.783E+01)≈	(1.435E+01)≈	(2.498E+01)
F12	4.004E+02	8.338E+02	9.806E+02	1.109E+03	1.279E+01	1.144E+03	2.609E+03	1.342E+03	4.920E+02	9.080E+02
	(2.218E+02)+	(3.660E+02)≈	(3.163E+02)≈	(4.253E+02)≈	(5.432E+01)+	(2.673E+02)−	(5.206E+03)−	(1.323E+03)≈	(2.284E+02)+	(4.222E+02)
F13	1.554E+01	1.559E+01	2.073E+01	3.629E+01	2.994E+01	2.130E+01	1.754E+02	2.061E+01	1.633E+01	2.197E+01
	(6.276E+00)+	(7.030E+00)+	(7.788E+00)≈	(1.407E+01)−	(1.665E+01)≈	(7.335E+00)≈	(6.135E+02)−	(7.349E+00)≈	(8.074E+00)+	(7.228E+00)
F14	2.273E+01	2.256E+01	2.544E+01	1.230E+03	1.139E+01	2.976E+01	2.862E+01	2.090E+01	2.049E+01	4.324E+01
	(1.066E+00)+	(1.425E+00)+	(2.396E+00)+	(2.441E+03)≈	(1.011E+01)+	(5.720E+00)+	(9.325E+00)≈	(9.585E+00)+	(1.002E+01)+	(2.566E+01)
F15	2.638E+00	4.037E+00	9.294E+00	1.117E+03	5.329E+00	7.911E+00	1.449E+01	6.056E+00	4.846E+00	2.959E+00
	(1.318E+00)≈	(1.793E+00)−	(2.940E+00)−	(3.082E+03)−	(2.920E+00)−	(4.976E+00)−	(1.119E+01)−	(2.991E+00)−	(1.873E+00)−	(1.592E+00)
F16	3.428E+01	5.368E+01	4.253E+02	2.464E+02	2.638E+02	5.874E+02	3.735E+02	2.329E+02	2.655E+02	8.593E+01
	(3.114E+01)≈	(6.389E+01)+	(1.645E+02)−	(1.575E+02)−	(2.033E+02)−	(2.510E+02)−	(2.114E+02)−	(1.319E+02)−	(2.489E+02)−	(1.318E+02)
F17	2.576E+01	4.155E+01	7.902E+01	4.042E+01	5.580E+01	8.209E+01	5.962E+01	3.008E+01	6.042E+01	3.819E+01
	(5.643E+00)≈	(9.161E+00)−	(4.747E+01)−	(3.005E+01)−	(9.446E+01)≈	(8.562E+01)−	(1.520E+01)−	(1.084E+01)≈	(3.751E+01)−	(4.258E+01)
F18	2.104E+01	2.169E+01	2.336E+01	1.607E+04	1.956E+01	2.571E+01	1.791E+02	2.469E+01	2.307E+01	5.895E+01
	(8.563E-01)+	(1.092E+00)+	(4.643E+00)+	(2.493E+04)−	(5.280E+00)+	(2.525E+00)+	(7.636E+02)≈	(4.075E+00)+	(5.644E+00)+	(4.071E+01)
F19	5.409E+00	7.531E+00	7.414E+00	2.961E+02	3.625E+00	6.359E+00	1.372E+01	6.164E+00	5.723E+00	7.818E+00
	(1.513E+00)+	(2.089E+00)≈	(2.700E+00)≈	(7.192E+02)−	(1.229E+00)+	(1.972E+00)+	(2.636E+00)−	(1.806E+00)+	(1.299E+00)+	(1.644E+00)
F20	2.977E+01	4.671E+01	8.695E+01	5.056E+01	1.456E+02	1.450E+02	9.660E+01	2.624E+01	7.588E+01	1.246E+02
	(3.949E+00)+	(1.787E+01)+	(5.236E+01)+	(5.831E+01)+	(3.133E+01)−	(1.063E+02)≈	(5.476E+01)≈	(1.262E+01)+	(7.730E+01)+	(3.878E+01)
F21	2.124E+02	1.955E+02	2.249E+02	2.236E+02	2.095E+02	2.379E+02	2.342E+02	2.225E+02	2.391E+02	2.076E+02
	(2.554E+00)−	(2.600E+01)+	(6.624E+00)−	(5.833E+00)−	(3.608E+00)−	(8.886E+00)−	(7.830E+00)−	(4.837E+00)−	(1.928E+01)−	(3.593E+00)
F22	1.000E+02	1.000E+02	2.012E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02
	(0.000E+00)≈	(1.154E-13)≈	(5.539E+02)≈	(0.000E+00)≈	(2.292E-13)≈	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)≈	(8.303E-14)
F23	3.569E+02	3.503E+02	3.814E+02	3.680E+02	3.532E+02	3.868E+02	3.763E+02	3.641E+02	3.769E+02	3.467E+02
	(3.216E+00)−	(3.275E+00)−	(8.612E+00)−	(8.167E+00)−	(8.438E+00)−	(9.786E+00)−	(1.116E+01)−	(5.939E+00)−	(9.196E+00)−	(6.419E+00)
F24	4.297E+02	4.261E+02	4.499E+02	4.409E+02	4.190E+02	4.602E+02	4.387E+02	4.388E+02	4.508E+02	4.186E+02
	(2.349E+00)−	(1.783E+00)−	(7.328E+00)−	(7.296E+00)−	(5.633E+00)≈	(1.324E+01)−	(8.561E+00)−	(4.882E+00)−	(1.204E+01)−	(4.897E+00)
F25	3.867E+02	3.865E+02	3.783E+02	3.870E+02	3.867E+02	3.868E+02	3.784E+02	3.869E+02	3.868E+02	3.867E+02
	(5.426E-03)≈	(8.147E-01)+	(2.912E-02)+	(1.913E-01)−	(2.754E-02)−	(5.720E-02)−	(1.137E-01)+	(8.992E-02)−	(5.742E-02)−	(9.805E-03)
F26	9.581E+02	4.380E+02	1.317E+03	1.139E+03	9.057E+02	1.310E+03	8.320E+02	1.096E+03	1.123E+03	8.131E+02
	(5.388E+01)−	(2.961E+02)+	(1.319E+02)−	(9.677E+01)−	(1.480E+02)≈	(2.460E+02)−	(4.837E+02)−	(8.144E+01)−	(3.016E+02)−	(2.377E+02)
F27	5.054E+02	5.027E+02	5.000E+02	5.024E+02	5.179E+02	5.036E+02	5.000E+02	4.990E+02	4.937E+02	5.106E+02
	(5.245E+00)+	(5.212E+00)+	(1.540E-04)+	(7.070E+00)+	(7.598E+00)−	(5.165E+00)+	(1.412E-04)+	(4.791E+00)+	(9.557E+00)+	(6.725E+00)
F28	3.107E+02	3.034E+02	4.985E+02	3.331E+02	3.241E+02	3.334E+02	3.275E+02	3.328E+02	3.230E+02	3.103E+02
	(3.264E+01)≈	(1.886E+01)≈	(3.953E+00)−	(5.154E+01)−	(4.443E+01)≈	(5.260E+01)−	(4.646E+01)−	(5.099E+01)−	(4.740E+01)≈	(3.152E+01)
F29	4.315E+02	4.350E+02	3.373E+02	4.222E+02	4.617E+02	4.613E+02	4.025E+02	4.413E+02	4.388E+02	4.263E+02
	(5.954E+00)−	(1.239E+01)−	(5.598E+01)+	(3.859E+01)≈	(4.293E+01)−	(4.424E+01)−	(6.040E+01)+	(1.883E+01)−	(3.317E+01)−	(3.038E+01)
F30	1.998E+03	2.004E+03	2.092E+02	2.097E+03	2.059E+03	2.033E+03	4.792E+02	2.021E+03	2.008E+03	1.985E+03
	(7.075E+01)≈	(7.070E+01)≈	(2.384E+00)+	(1.180E+02)−	(4.010E+01)−	(6.830E+01)−	(6.847E+02)+	(5.096E+01)−	(5.898E+01)≈	(3.326E+01)
−	11	10	15	20	13	19	17	16	17	
+	10	12	9	3	7	6	7	6	8	
≈	9	8	6	7	10	5	6	8	5	

According to Table 7 for the functions with 30 in dimensionality, our HS-ES-DE performs significantly better than L-SHADE-EpSin, UMOEAs-II, MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, and NDE on 11, ten, 15, 20, 13, 19, 17, 16, and 17 functions, respectively. Meanwhile, HS-ES-DE is inferior to the peers on ten, 12, nine, three, seven, six, seven, six, and eight functions, respectively. That is, although our algorithm loses to UMOEAs-II, it has better performance than all of the other peers. More details are given below.

- For the three unimodal functions, HS-ES-DE is superior to UMOEAs-II on two functions (F1 and F2), MPEDE on two functions (F2 and F3), ETI-JADE on one function (F2), HS-ES on three functions (F1-F3), AMECoDEs on one function (F2), EDEV on one function (F2), MLCC-SI on two functions (F2 and F3), and NDE on two functions (F2 and F3). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on two functions (F1 and F3), AMECoDEs on one function (F3), EDEV on one function (F1) and NDE on one function (F1). That is, our algorithm performs better than UMOEAs-II, MPEDE, ETI-JADE, HS-ES, MLCC-SI, and NDE, but performs worse than L-SHADE-EpSin. Meanwhile, our algorithm is comparable to AMECoDEs and EDEV.

- For the seven simple multimodal functions, HS-ES-DE is superior to L-SHADE-EpSin on six functions (F4-F8, and F10), UMOEAs-II on three functions (F4, F6, and F10), MPEDE on five functions (F5-F8, and F10), ETI-JADE on five functions (F4, F5, F7, F8, and F10), HS-ES on one function (F10), AMECoDEs on six functions (F4-F8, and F10), EDEV on four functions (F5, F7, F8, and F10), MLCC-SI on four functions (F5, F7, F8, and F10), and NDE on six functions (F4-F8, and F10). Meanwhile, our algorithm is inferior to UMOEAs-II on three functions (F5, F7, and F8), MPEDE on one function (F4), ETI-JADE on one function (F6), HS-ES on two functions (F4 and F6), EDEV on tWO functions (F4 and F6), MLCC-SI on one function (F6). That is, our algorithm performs better than L-SHADE-EpSin, MPEDE, ETI-JADE, AMECoDEs, EDEV, MLCC-SI, NDE, but performs worse than HS-ES. Meanwhile, our algorithm is comparable to UMOEAs-II.

- For the ten hybrid functions, HS-ES-DE is superior to UMOEAs-II on two functions (F15 and F17), MPEDE on three functions (F15-F17), ETI-JADE on seven functions (F11, F13, F15-F19), HS-ES on three functions (F15, F16, and F20), AMECoDEs on four functions (F12, and F15-F17), EDEV on seven functions (F11-F13, F15-F15, and F19), MLCC-SI on two functions

(F15 and F16), and NDE on three functions (F15-F17). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on seven functions (F11-F14, and F18-F20), UMOEAs-II on five functions (F13, F14, F16, F18, and F20), MPEDE on four functions (F11, F14, F18, F20), ETI-JADE on one function (F20), HS-ES on five functions (F11, F12, F14, F18 and F19), AMECoDEs on four functions (F11, F14, F18, and F19), MLCC-SI on four functions (F14 and F18-F20), and NDE on six functions (F12-F14 and F18-F20). That is, our algorithm performs better than ETI-JADE, EDEV, but performs worse than L-SHADE-EpSin, UMOEAs-II, MPEDE, HS-ES, MLCC-SI, and NDE. Meanwhile, our algorithm is comparable to AMECoDEs.

- For the eight composition functions, HS-ES-DE is superior to L-SHADE-EpSin on five functions (F21, F23, F24, F26, and F29) UMOEAs-II on three functions (F23, F24, and F29), MPEDE on five functions (F21, F23, F24, F26 and F28), ETI-JADE on seven functions (F21, F23-F26, F28, and F30), HS-ES on six functions (F21, F23, F25, F27, F29, and F30), AMECoDEs on eight functions (F21, F23-F26, and F28-F30), EDEV on five functions (F21, F23, F24, F26, and F28), MLCC-SI on eight functions (F21, F23-F26, and F28-F30), and NDE on six functions (F21, F23-F26, and F29). Meanwhile, our algorithm is inferior to L-SHADE-EpSin on one function (F27), UMOEAs-II on four functions (F21, and F25-F27), MPEDE on four functions (F25, F27, F29 and F30), ETI-JADE on one function (F27), AMECoDEs on one function (F27), EDEV on four functions (F25, F27, F29 and F30), MLCC-SI on one function (F27), and NDE on one function (27). That is, our algorithm performs better than L-SHADE-EpSin, MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, and NDE, but performs worse than UMOEAs-II.

The Friedman test for this experiment is given in Table 8. It can be seen that our algorithm ranks third.

4.4. Comparison based on the CEC 2020 suite

In this experiment, our algorithm is compared with the top three populaiton-based metaheuristics in the 2020 competition, IMODE, AGSK, and j2020. As mentioned before, both the value of $MaxFES$ taken in the previous CEC competitions and that in the 2020 competitions are employed here. The experimental results for the CEC 2020 functions with 20 in D are listed in Table 9. Note that dimensionality can be just set 10, 15, and 20 in the new suite.

Table 8: The Friedman test for the second experiment based on the CEC 2017 benchmark test suite

Algorithm	Ranking	Algorithm	Ranking
L-SHADE-Epsin	3.483	AMECoDEs	7.567
UMOEAs-II	4.000	EDEV	6.583
MPEDE	6.333	MLCC-SI	5.233
ETI-JADE	6.800	NDE	6.017
HS-ES	4.783	HS-ES-DE	4.200

Table 9: Results of the ten algorithms for the CEC 2020 functions with 20 in dimensionality. Here, old rule means $MaxFES = 2.0E+05$, while new rule means $MaxFES = 1.0E+07$. "+" or "−" denotes that the current result is significantly better or statistical worse than the result of our algorithm in terms of Wilcoxon's rank sum test at a 0.05 significance level, respectively. Meanwhile, "≈" represents that there is no significant difference

Function	Average (standard deviation)							
	IMODE		AGSK		j2020		HS-ES-DE	
	Old rule	New rule	Old rule	New rule	Old rule	New rule	Old rule	New rule
F1	1.503E-04	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	(1.870E-05)−	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)≈	(0.000E+00)	(0.000E+00)
F2	2.504E+02	3.463E-01	3.687E+02	1.233E+00	1.350E+02	6.246E-02	2.150E+01	1.767E+01
	(1.138E+02)−	(5.663E-01)+	(6.486E+01)−	(1.359E+00)≈	(9.986E+01)−	(0.000E+00)+	(5.200E+01)	(4.153E+01)
F3	3.295E+01	2.046E+01	3.179E+01	2.039E+01	2.700E+01	0.000E+00	2.140E+01	2.071E+01
	(3.820E+00)−	(1.245E-01)+	(1.627E+00)−	(0.000E+00)+	(2.647E+01)−	(0.000E+00)+	(6.664E-01)	(2.614E-01)
F4	7.171E-02	4.995E-01	1.933E+00	7.749E-01	1.408E+00	1.135E-01	0.000E+00	0.000E+00
	(5.441E-02)−	(8.077E-02)−	(1.985E-01)−	(1.006E-01)−	(1.464E+00)−	(8.475E-02)−	(0.000E+00)	(0.000E+00)
F5	9.493E+02	1.055E+01	3.503E+02	2.917E+01	6.239E+02	1.461E+02	2.984E+02	2.667E+02
	(3.531E+02)−	(4.112E+00)+	(1.265E+02)≈	(2.939E+01)+	(2.793E+02)−	(7.067E+01)+	(1.565E+02)	(1.776E+02)
F6	2.505E+00	2.868E-01	2.117E+00	1.605E-01	1.142E+00	1.965E-01	5.317E+00	9.831E-01
	(5.023E-01)+	(7.395E-02)+	(2.597E-01)+	(4.994E-02)+	(1.204E+00)+	(1.117E-01)+	(2.175E+01)	(4.973E-01)
F7	5.886E+02	5.098E-01	7.295E+01	5.664E-01	1.406E+02	3.961E-01	1.898E+02	1.441E+02
	(2.119E+02)−	(1.780E-01)+	(5.262E+01)≈	(7.354E-01)+	(1.435E+02)−	(1.711E+01)+	(2.143E+02)	(1.243E+02)
F8	1.004E+02	8.423E+01	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02	1.000E+02
	(6.515E-01)−	(1.833E+01)+	(3.700E-13)≈	(8.303E-14)≈	(1.000E+02)≈	(1.000E+02)+	(0.000E+00)	(8.303E-14)
F9	1.067E+02	9.667E+01	4.047E+02	9.667E+01	4.225E+02	1.000E+02	3.940E+02	3.891E+02
	(2.568E+01)+	(1.826E+01)+	(5.759E+01)−	(1.826E+01)+	(4.338E+02)+	(4.113E+02)+	(5.021E+00)	(1.806E+01)
F10	4.196E+02	3.997E+02	4.137E+02	3.994E+02	4.138E+02	3.990E+02	4.125E+02	4.126E+02
	(1.674E+01)−	(4.441E-01)+	(8.436E-03)−	(2.008E+00)+	(4.137E+02)−	(3.991E+02)+	(1.361E+00)	(1.475E+00)
−	8	1	5	1	6	1		
+	2	8	1	6	2	8		
≈	0	1	4	3	2	1		

According to Table 9, when $MaxFES = 2.0E{+}05$, our HS-ES-DE performs significantly better than IMODE, AGSK, and j2020 on eight, five, and six functions, respectively. Meanwhile, HS-ES-DE is inferior to the peers on two, one, and two function(s), respectively. That is, our algorithm has better performance than all of the peers. More details are given below.

- For the only one unimodal function, HS-ES-DE is superior to IMODE.
- For the three simple multimodal functions, HS-ES-DE is superior to all the peers on all the function.
- For the three hybrid functions, HS-ES-DE is superior to IMODE on two functions (F5 and F7), and j2020 on two functions (F5 and F7). Meanwhile, our algorithm is inferior to IMODE on one function (F6), AGSK on one function (F6), and j2020 on one function (F6). That is, our algorithm performs better than IMODE and j2020, but performs worse than AGSK.
- For the three composition functions, HS-ES-DE is superior to IMODE on two functions (F8 and F10), AGSK on two functions (F9 and F10), and j2020 on one function (F10). Meanwhile, our algorithm is inferior to IMODE on one function (F9), j2020 on one function (F9) That is, our algorithm performs better than IMODE and AGSK. Meanwhile, our algorithm is comparable to j2020.

When $MaxFES = 1.0E{+}07$, our HS-ES-DE performs significantly better than IMODE, AGSK, and j2020 on one, one, and one function, respectively. Meanwhile, HS-ES-DE is inferior to the peers on eight, six, and eight functions, respectively. That is, the performance of our algorithm is worse than all of the peers.

Table 10: The Friedman test for the experiment based on the CEC 2020 benchmark test suite

Algorithm	Old rule Ranking	New rule Ranking
IMODE	3.300	2.200
AGSK	2.500	2.400
j2020	2.400	1.950
HS-ES-DE	1.800	3.450

4.5. Result analysis

It can be seen that, based on the old rule on $MaxFES$, our algorithm is very competitive in comparison with all the peers - L-SHADE-EpSin, UMOEAs-II, MPEDE, ETI-JADE, HS-ES, AMECoDEs, EDEV, MLCC-SI, NDE, IMODE, AGSK, and j2020. However, based on the new rule on $MaxFES$, our HS-ES-DE performs much worse than the top three algorithms in the 2020 competition, IMODE, AGSK, and j2020. In fact, the old rule is for comparing solving ability during a short course, while the new one is for comparing that during a long course. In summary, our HS-ES-DE has a sound solving ability during a short course.

5. Conclusion

For years, population-based metaheuristics have been used for real parameter single objective optimization. The competitions on this topic have been held in the series of CEC. Based on L-SHADE-EpSin and HS-ES, two winners in the competitions, we propose ensemble with successively executed constituent algorithms - HS-ES-DE. In our ensemble, HS-ES is executed firstly. As soon as stagnation is detected by our scheme, L-SHADE-EpSin takes over population after our specific processing on population. The experiment results show that our ensemble is competitive among population-based metaheuristics for real parameter single objective optimization.

Now that taking over in a right occasion can bring improvement in solution, taking over alternately deserves of study. In our future study, we will try to enhance HS-ES-DE. More times of taking over may occur in the new version. That is, L-SHADE-EpSin may take over population from HS-DE, and vice versa. To this end, processing on population for taking over becomes more complex than before. Such a new version of HS-ES-DE may perform better than the current one and be fit for either a short course or a long course.

References

[1] Mostafa Z Ali, Noor H Awad, Ponnuthurai Nagaratnam Suganthan, and Robert G Reynolds. An adaptive multipopulation differential evolution with dynamic population reduction. *IEEE Trans. on Cybernetics*, 47(9):2768–2779, 2017.

[2] Noor H Awad, Mostafa Z Ali, Ponnuthurai N Suganthan, and Robert G Reynolds. An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In *Proc of CEC*, pages 2958–2965. IEEE, 2016.

[3] Laizhong Cui, Genghui Li, Zexuan Zhu, Qiuzhen Lin, Ka-Chun Wong, Jianyong Chen, Nan Lu, and Jian Lu. Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism. *Information Sciences*, 422:122–143, 2018.

[4] Wei Du, Sunney Yung Sun Leung, Yang Tang, and Athanasios V Vasilakos. Differential evolution with event-triggered impulsive control. *IEEE Trans. on Cybernetics*, 47(1):244–257, 2017.

[5] Saber Elsayed, Noha Hamza, and Ruhul Sarker. Testing united multi-operator evolutionary algorithms-II on single objective optimization problems. In *Proc of CEC*, pages 2966–2973. IEEE, 2016.

[6] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, 11(1):1–18, 2003.

[7] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[8] Abhishek Kumar, Rakesh Kumar Misra, and Devender Singh. Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In *Proc of CEC*, pages 1835–1842. IEEE, 2017.

[9] Chuang Liu and Linan Fan. A hybrid evolutionary algorithm based on tissue membrane systems and cma-es for solving numerical optimization problems. *Knowledge-Based Systems*, 105:38–47, 2016.

[10] Ilya Loshchilov. CMA-ES with restarts for solving CEC 2013 benchmark problems. In *Proc of CEC*, pages 369–376. IEEE, 2013.

[11] Ali W Mohamed, Anas A Hadi, Anas M Fattouh, and Kamal M Jambi. Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems. In *2017 IEEE Congress on evolutionary computation (CEC)*, pages 145–152. IEEE, 2017.

[12] Karam M Sallam, Saber M Elsayed, Ripon K Chakrabortty, and Michael J Ryan. Improved multi-operator differential evolution algorithm for solving unconstrained problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020.

[13] Karam M Sallam, Saber M Elsayed, Ruhul A Sarker, and Daryl L Essam. Landscape-based adaptive operator selection mechanism for differential evolution. *Information Sciences*, 418:383–404, 2017.

[14] Urban Škvorc, Tome Eftimov, and Peter Korošec. CEC real-parameter optimization competitions: Progress from 2013 to 2018. In *Proc of CEC*, pages 3126–3133. IEEE, 2019.

[15] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[16] Ryoji Tanabe and Alex Fukunaga. Success-history based parameter adaptation for differential evolution. In *Proc of CEC*, pages 71–78. IEEE, 2013.

[17] Ryoji Tanabe and Alex S Fukunaga. Improving the search performance of SHADE using linear population size reduction. In *Proc of CEC*, pages 1658–1665. IEEE, 2014.

[18] Mengnan Tian and Xingbao Gao. Differential evolution with neighborhood-based adaptive evolution mechanism for numerical optimization. *Information Sciences*, 478:422–448, 2019.

[19] Yong Wang, Zixing Cai, and Qingfu Zhang. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans. on Evo. Comput.*, 15(1):55–66, 2011.

[20] Guohua Wu, Rammohan Mallipeddi, Ponnuthurai Nagaratnam Suganthan, Rui Wang, and Huangke Chen. Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences*, 329:329–345, 2016.

[21] Guohua Wu, Xin Shen, Haifeng Li, Huangke Chen, Anping Lin, and PN Suganthan. Ensemble of differential evolution variants. *Information Sciences*, 423:172–186, 2018.

[22] Geng Zhang and Yuhui Shi. Hybrid sampling evolution strategy for solving single objective bound constrained problems. In *Proc of CEC*, pages 1–7. IEEE, 2018.

[23] Jingqiao Zhang and Arthur C Sanderson. JADE: adaptive differential evolution with optional external archive. *IEEE Trans. on Evo. Comput.*, 13(5):945–958, 2009.

[24] Sheng Xin Zhang, Li Ming Zheng, Kit Sang Tang, Shao Yong Zheng, and Wing Shing Chan. Multi-layer competitive-cooperative framework for performance enhancement of differential evolution. *Information Sciences*, 482:86–104, 2019.